BIT-LET

öltet SZÁMÍTÁSTECHNIKAI MELLÉKLET · 1987 NOVEMBER 50



Olvasom az újságot. Az újságban pedig megírták, hogy rövidesen újabb kísérletet kezdenek Magyarországon. A kísérletre kiszemelt városka Eger. A kísérlet tárgya pedig a mikroelektronika nem is olyan új vívmánya, az információt tároló hitelkártya, amellyel odamehet tulajdonosa az utcán felszerelt automatához, s bármely napszakban pénzt vehet ki számlájáról, s amellyel lehetőség szerint minél több helyen – üzletben, étteremben, OTP-fiókban – fizethet is. A hír voltaképpen örvendetes, hiszen arról tanúskodik, hogy a technika eme áldásos vívmánya lassanként hozzánk is betör. Elmaradásunk a területen nem nagyobb, mint sok másban. Nyugaton turistáskodó honfitársaink látásból már jól ismerik ezt a technikai vívmányt, hiszen sok éve, hogy először csodálhatták meg az efféle automatákat Jugoszláviától Amerikáig.

A hírnek azt a részét nem értem csupán, hogy mi ebben a kísérlet. Hiszen egy sokszor és sok helyen kipróbált dologról van szó.

A rendszer működőképes, bevált, bizonyíték éppen elég akad rá.

Pénzünk arra, hogy bevezessük az egész országban nyilván annál kevesebb. Ez idáig világos. Igény a bevezetésére minimális. Legalábbis erre kell következtetnünk abból, hogy az OTP-takarékcsekk nevű „műintézmény" iránt sem egetverő az érdeklődés. Márpedig e pillanatban ez az egyetlen olyan fizetőeszköz kis hazánkban, amely bárki számára hozzáférhető úgynevezett pénzkímélő lehetőség. E sorok írója mint a modern dolgok iránt beoltott érdeklődő kipróbálta már ezt a csekket. S ily módon megmondhatója, hogy az állampolgárok érdektelensége egyszerű okkal magyarázható. Azzal ugyanis, hogy az OTP-takarékcsekk csak arra jó, hogy gazdáját is, embertársait is fölbosszantsa. Áll az ember a harmincméteres sorban például a Skála élelmiszer osztályának pénztáránál. Sorra kerül. A számla 1200 forintot mutat. Az ember előveszi csekkfüzetét, kitölt egy csekket. Előhalássza a személyijét, s odaadja a pénztárosnak. A pénztáros megkövülten bámul az emberre. Némi habozás után segítségért kiált. Gizike, az ügyeletes üzletvezető 2–3 perc után előkerül. Ő már tudja, hogy miről van szó, hát elmagyarázza a pénztárosnőnek. Valami nyomtatvány kerül elő, ezt kitöltik... És ez így megy kb. 3–4 percig. Közben a mögöttem álló sorban állókat megüti a guta. Nekem meg égjen le a pofámról a bőr, hogy egyáltalán ilyesmit merészelek.

Hát így. Így megy abban a kevés üzletben, ahol egyáltalán elfogadják a csekket. Mert a legtöbb helyen el sem fogadják. Így azután az állampolgár enyhén szólva idegenkedik a modern kor eme vívmányaitól. Nem csoda tehát ha kísérletezünk. Kísérletezünk olyan dolgokkal, amelyek sokszorosan ki vannak találva. Nem nehéz megjósolni az egri kísérlet eredményét. A bátor, vállalkozó kedvű egriek (minthogy szülővárosomról van szó – különösen szurkolok nekik) közül néhány ezren nyilván élnek majd a felkínált lehetőséggel. Kiváltják kis kártyájukat, s néhány hónap múltán megüti őket a guta. Kártyájuk ugyanis csak bonyodalmat okoz majd nekik. Lesznek azután a városban „esetek". Nyilván lesznek vandál elemek, akik némi készpénz reményében megpróbálnak fölfeszíteni, szétverni egy-egy automatát. Eredményük csak annyi lesz, hogy néhány hétre elromlik a szerkezet. Cikkek jelennek majd meg a Heves megyei Népújságban, amelyekben jól leteremtik az egri népet, hogy nem elég érett még egy ilyen kísérletre. Ellencikkek is születnek majd, amelyek megírják, hogy nem a nép, hanem a technika, meg az egész infrastruktúra a hibás. Én már előre ezeknek az ellencikkeknek a szerzőivel értek egyet. Úgy gondolom ugyanis, hogy nem kísérletezni kellene, hanem a bevált receptek szerint föl kellene építeni egy rendszert. Olyat, amelyről bebizonyosodott, hogy működik. Ez az egész ugyanis akkor ér valamit, ha minél szélesebb körben használhatja tulajdonosa a kártyát. Az automata szétverések, feszegetések pedig technikailag fejlettebb országokban sem számítottak ritkaságnak – eleinte. Azután ezeket a berendezéseket is megszokták.

És ez lenne a lényeg. Megszokni, hétköznapivá tenni az új technika áldásait. Ami működik, abban ugyanis bíznak az emberek. Tessék például a kerékre gondolni. Vagy lehet, hogy nálunk azt is először kísérleti jelleggel vezették be?

Angyalosi László

HÍROLDAL



ZÖVEGFORDÍTÓ

A British Telecom telekommunikációs óriásvállalat bemutatta szinkronfordításra alkalmas új számítógépét. A berendezés angolról franciára, illetve franciáról angolra fordít, de kapacitása egyelőre igen korlátozott. Csak mintegy 100 szavas szókészlettel rendelkezik, bár ennél több szabványkifejezést tolmácsolhat egy-egy telefonbeszélgetésben. A British Telecom kutatási igazgatója szerint a számítógép fontos technikai áttörést jelent. A társaság reméli, hogy a számítógép továbbfejlesztett változata öt éven belül átveheti a szikrontolmácsok szerepét.

SZOFTVERVÉDELEM

Japánban Software Information Center-t alakítottak ki, amelyben minden hazai fejlesztésű szoftvert nyilvántartanak és 50 évig tárolnak. Ez idő alatt e központ a szabadalmi jognak megfelelő iparjogvédelmet biztosít a szoftver kifejlesztőinek. Az állam a japán szoftveripar fejlesztése érdekében hívta életre a központot. Az importált szoftvereket is nyilvántartják és összehasonlítják azokat a hasonló célú hazai termékekkel. Az USA-ból származó szoftvereket ugyanolyan iparjogvédelemben részesítik, mint a japánokat, de ez ideig tisztázatlan, hogy a védelem kiterjed-e az európai származású szoftverekre is?

MUNKAFELÜGYELŐK

Egy washingtoni jelentés szerint már több mint hétmillió amerikai állampolgár munkáját figyelik, ellenőrzik számítógépek. Illetékesek valószínűsítik, hogy a gépi ellenőrök okozta stressz tovább növeli a munkaártalmakból származó megbetegedéseket. Egy tanulmány szerint a számítógép könyörtelen és tökéletes munkafelügyelő: például számontartja, hogy hány betűt ütött le a gépírónő, abból hányat tévesen, hányszor telefonált, hány rossz levelet dobott a szemétkosárba és hányszor hagyta el munkahelyét.

ELMEZAVARÓ GÉP

Az elmeosztályra került egy tizennyolc éves dán fiatalember, személyi számítógépéhez kötődő mániája miatt. Az ifjú számítógéprajongó, aki már iskolába sem volt hajlandó járni, napi tizenhat órát töltött el gépével, mindent elhanyagolt, csak számítógépnyelven volt képes gondolatait kifejezni és álmában is programokat futtatott. Barátjának csak a számítógépet tekintette, és környezetére úgy nézett, mint félresikerült áramköri elemek sokaságára.

-CLN

THOMSON–VIDEOTON

Korszerű mikroprocesszoros televíziócsalád sorozatgyártásának megkezdésére készül a székesfehérvári Videoton Elektronikai Vállalat. A gyártási technológiát a francia Thomson cégtől vásárolta, a gyártásra kerülő készüléket pedig – figyelembe véve az itthoni igényeket, valamint a hazai üzemekben és a szocialista országokban előállított alkatrészek felhasználásának lehetőségét is – a két vállalat szakemberei közösen fejlesztették ki. A tetszetős külsejű, korszerű belső felépítésű, műszakilag igen megbízható készülékek mikroprocesszor programját úgy alakították ki, hogy kisebb módosításokkal többféle szabvány szerint is működtethetők, illetve bármely adóra ráhangolhatók legyenek. Az exportképességet az is növeli, hogy a készülékek továbbfejlesztésével, illetve megfelelő berendezések beiktatásával alkalmasak lesznek a műholdas adások vételére is. Az új készülékek sorozatgyártása 1988 második felében indul.

SZUPERMEMÓRIA

Mintegy kettőszázötven gépelt oldalnyi szöveg tárolására képes az új szupermemória-lapka, amelyet a Siemens, a holland Philips és az NSZK-beli Valvo közösen fejlesztett ki. Erre a célra újfajta háromdimenziós memóriacellát használnak, amelyben kilencvenegy négyzetmilliméteren 8,6 millió alkatrész funkcióját tudják elhelyezni. Az új félvezető cella összesen négymillió bit tárolására alkalmas.

IBM CHIP

Az amerikai IBM cég bejelentette, hogy létrehozta 4 megabites chipjének prototípusát. A ma használatban lévő legnagyobb chipek kapacitása 1 megabit. A 4 megabit körülbelül annyit jelent, hogy egy lapka négyszáz gépelt oldalnyi információt képes tárolni. Az új IBM-chip bejelentés érdekessége, hogy a 4 megás változatot is a régi – az 1 megabitesek gyártásához használt – gyártósoron állították elő. Az új chip alig 35%-kal nagyobb, mint az 1 megás, és kevesebbet is fogyaszt annál, gyorsasága pedig kiemelkedő: a memóriarekeszből egy bitnyi információt mindössze 65 milliomod másodperc alatt elő lehet hívni.

ORIENT EXPRESS

Az Egyesült Államokban egy új, az eddigi típusoknál sokkalta nagyobb teljesítményű számítógépet fejlesztettek ki. A 100 millió dollár értékű computert a NASA, az amerikai repülési és űrhajózási hivatal avatta fel. A repülés tudósai szerint az új űrhajó- és repülőgéptípusok megépítése a számítástechnika fejlődésétől függ. Ami a gyakorlati alkalmazást illeti, a tudósok már dolgoznak a következő évszázad repülőgépének kifejlesztésén, várhatóan 10 éven belül meglesznek vele. „munkaneve" jelenleg Orient Express. Hiperszonikus – 25-szörös hangsebességű –, s csupán két óra kell, hogy az Egyesült Államokból Japánba repüljön úgy, hogy közben az űrbe emelkedik.

ÖREG FÁK

A szombathelyi Berzsenyi Dániel Tanárképző Főiskola természetbarát oktatói és hallgatói elhatározták, hogy feltérképezik a Vas megyében található értékes, öreg fákat. A munkában segítenek a megye búvárklubjai, természetjáró csoportjai és az úttörők. Nemcsak a fák adatait írják össze, hanem az esetleg hozzájuk fűződő történeteket is. Az egybegyűlt anyagot a főiskola a KISZ környezetvédő klubjának tagjai számítógépen dolgozzák fel, térképen jelölik be az egyes fákat, facsoportokat, s programot készítenek természeti értékek megóvására, gondozására.

KÍMÉLŐ ÉTELEK

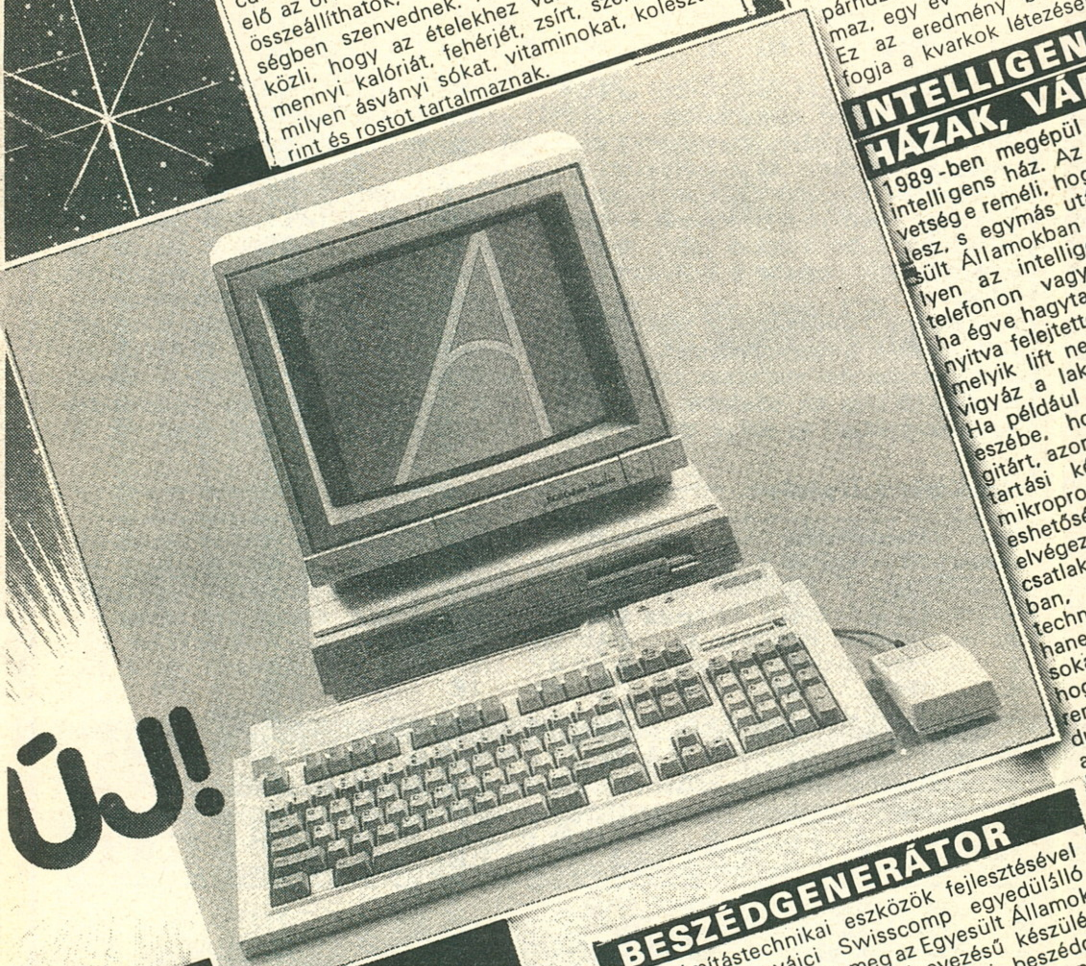
A kórházakban fekvő betegek speciális
tápanyagigényének megfelelő, több mint hét-
száz ételreceptet tartalmazó számítógépes
programot készítettek az MTA SZTAKI szak-
emberei, az Országos Diabetikai Intézet,
valamint az Országos Testnevelési és Sport-
egészségügyi Intézet kutatóival együttmű-
ködve. Az étlapon levesek, levesbetétek, fő-
zelékek, sültek, köretek és tészták szerepel-
nek. A program egy-egy étel készítésének
többféle változatát is megadja, annak meg-
felelően, hogy az egyes betegségeknek só-
és zsírszegény, tej-, liszt-, nátrium- vagy
cukormentes, illetve fehérjegazdag ételt ír
elő az orvos. Az ételek olyanoknak is rögtön
összeállíthatók, akik egyszerre több beteg-
ségben szenvednek. A számítógép írásban
közli, hogy az ételekhez való alapanyagok
mennyi kalóriát, fehérjét, zsírt, szénhidrátot,
milyen ásványi sókat, vitaminokat, koleszte-
rint és rostot tartalmaznak.

KVARKOK

Nem vezettek eredményre a kvarkok létezését
kutató kísérletek, amelyeknek azt kellett
volna bizonyítaniuk, hogy a protonok és a
neutronok kvarkokból és antikvarkokból áll-
nak. Ezért az elméleti fizikusok egy IBM szu-
perszámítógéppel eredtek a probléma nyo-
mába. A kvantumelmélet egyenleteinek meg-
oldására eddig nem állt rendelkezésre olyan
számítógép, amely a modellezés után száz
éven belül választ adott volna a kérdésekre.
Az IBM GF–11-es típusú gépe, amely 576
párhuzamosan kapcsolt processzort tartal-
maz, egy év múlva megoldást szolgáltathat.
Ez az eredmény bizonyítani vagy cáfolni
fogja a kvarkok létezését.

INTELLIGENS HÁZAK, VÁROSOK

1989-ben megépül New York-ban az első
intelligens ház. Az Amerikai Építészek Szö-
vetsége reméli, hogy a mintaépületnek sikere
lesz, s egymás után épülnek majd az Egye-
sült Államokban az ilyen házak. Vajon mi-
lyen az intelligens ház? A lakót például
telefonon vagy hangszórón figyelmezteti,
ha égve hagyta a gázt vagy a világítást, vagy
nyitva felejtette a lakás ajtaját, vagy ha vala-
melyik lift nem működik. A kis számítógép
vigyáz a lakóra, akármilyen szórakozott is.
Ha például egy fiúnak a fürdőszobában jut
eszébe, hogy bekapcsolja az elektromos
gitárt, azonnal megszakítja az áramot. A ház-
tartási készülékekbe vagy felszerelésekbe
mikroprocesszort építenek be, és a minden
eshetőséget bölcsen számba vevő program
elvégezteti a munkát. Minden elektromos
csatlakozó felhasználható bármire. Japán-
ban, ahol rendkívül fogékonyak minden
technikai újítás iránt, már nemcsak házakat,
hanem intelligens városrészeket vagy váro-
sokat is akarnak tervezni. Arra gondolnak,
hogy az új épületek lakóit egy számítógépes
rendszer arról is értesíti, hol van forgalmi
dugó, üres parkolóhely vagy olyan áruház,
amelyben abban az órában éppen nem to-
longanak a vevők. A japán építésügyi mi-
nisztérium 22 várost kíván „intelligens”
városnak minősíteni. Ezekben a városokban
olyan lakótelepek épülnek majd, melyekben
megszűnik a zsúfoltság. Az intelligens város-
részek lakói ha nem sajnálják a költséget,
bekapcsolódhatnak információs műhold-
programokba is. Kotaro Kitamura, a japán
építésügyi minisztérium főosztályvezetője
szerint a 2000. évben Japánnak 130 millió
lakosa lesz, s hetven százalékuk „intelli-
gens” városrészek „okos” házaiban él majd.



ÚJ!

A 310 ARCHIMEDES

**Az angol Acorn cég új terméke az
Archimedes A 310 a Byte szaklap több,
működési jellegzetességre vonatkozó
benchmark vizsgálata alapján, talán a
világ leggyorsabb személyi számítógé-
pe, amely másodpercenként átlagosan
négymillió műveletet végez. RAM-ja 1
Mbyte-os. 800 Kbyte-os, 3,5 inches
floppy meghajtót építettek bele. Az
A 310 egy újonnan szabadalmaztatott,
Arthur nevezetű operációs rendszer
alatt fut, amely részben kompatíbilis a
BBC Microval, és egy 512 Kbyte-os
ROM-ban van elhelyezve. Az Archime-
des erősen támogatja a színes grafikát.
Több mint 256 szín jeleníthető meg a
képernyőn. A rendszer nyolc sztereo
hangcsatornával működik. Az A 310-hez
egy hárombillentyűs egér kapcsolódik.**

BESZÉDGENERÁTOR

A számítástechnikai eszközök fejlesztésével
foglalkozó, svájci Swisscomp egyedülálló
berendezéssel jelent meg az Egyesült Államok
piacán. Smart Speaker elnevezésű készülé-
kük bármilyen szöveget emberi beszéddé
alakít át. Kifejlesztői szerint ez a világon a
legintelligensebb ilyen típusú készülék, amely
a legkülönbözőbb típusú számítógépekhez
illeszthető.

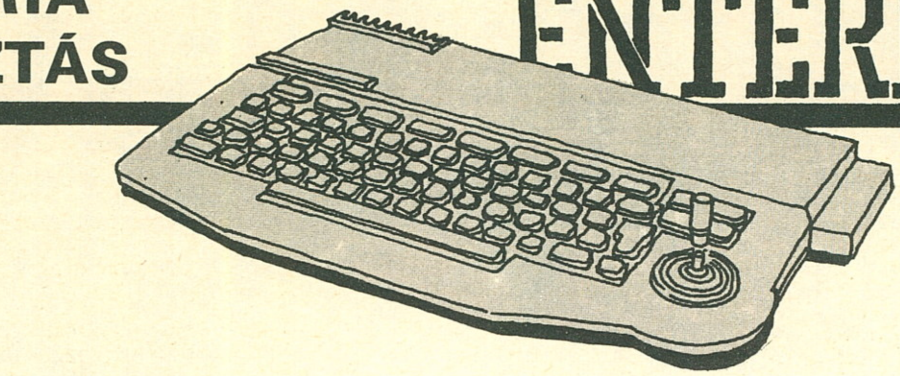
KRESZ

Másfél évvel ezelőtt az Országos Közlekedés-
biztonsági Tanács az International House
Budapesttel közösen pályázatot hirdetett az
általános iskolák Kresz-oktatását segítő szá-
mítástechnikai oktató és játékprogramok
megalkotására. A kezdeti lanyhább érdeklő-
dés és néhány technikai probléma volt az
oka annak, hogy a pályadíjas program elég
nehezen készült el. De a lényeg az, hogy ma
már a forgalmazásra, értékesítésre vállalkozó
Novotrade szaküzletében a négy részből
álló oktató szettet kilencszáznyolcvan fo-
rintért megvásárolhatják az érdeklődők. For-
galomba kerül – és külön is megvehető – a
programhoz tartozó játékkazetta is, ennél va-
lamivel olcsóbb áron.

ötlet

MEMÓRIA FELOSZTÁS

ENTERPRISE®



Az ENTERPRISE számítógép 128 Kbyte RAM-mal és 32–32 Kbyte belső, ill. külső ROM-mal rendelkezik. A gép Z–80 típusú mikroprocesszora azonban csak 64 K memóriát tud közvetlenül megcímezni. A tervezők ezt egy ún. lapozó logika beépítésével oldották meg, amely a külső fizikai memóriát (ez max. 4 Mbyte lehet) és a Z–80 64 K logikai (látszólagos) memóriáját egyaránt 16 Kbyte-os ún. lapokra osztották fel. A Z–80 P 4 db logikai lapjának brmelyikéhez hozzárendelhetjük a max. 256 fizikai memória lap bármelyikét. A hozzárendelést 4 regiszterrel végezzük el, amely perifériaként lett kialakítva.

A kiosztást az alábbi ábra mutatja:

Z–80 cím	Port cím
I 0000–3FFF	176=B0H
II 4000–7FFF	177=B1H
III 8000–BFFF	178=B2H
IV C000–FFFF	179=B3H

A portok írhatók és olvashatók is.

A fizikai memória felosztása:

Lapszám	Tartalom
248–255	RAM
0 és 2	Beépített 32K
1 és 3	ROM
4 és 6	Külső 32K
5 és 7	ROM

A rendszerváltozókat a program a 255. lapon helyezi el, a BASIC programot a 248. laptól kezdi elhelyezni. A 248. lapon van a megszakításrutin, valamint az RST 30 rutin belépési pontja, tehát ez a lap van a Z–80 I. lapján.

A billentyűzet

Sokszor szükség lehet arra, hogy több billentyű egyszerre történő lenyomását figyeljük, ill. gépi kódból olvassuk le a billentyűzetet. A billentyűzetet a gépen szokás szerint mátrixba kötötték. A mátrix leolvasása a 181. I/O porton keresztül történik. Először ki kell adni a 181. portra a leolvasni kívánt sor számát, majd ugyanezen a porton leolvashatjuk az adott sorban levő billentyű állapotát. Egy billentyű lenyomása esetén a hozzá tartozó bit 0 lesz. Ha BASIC-ből akarjuk leolvasni, akkor egyszerűbb mód is van, a megszakításrutin ugyanis leolvassa a billentyűt, és a leolvasott értékeket tárolja a memóriában. Itt azonban egy billentyű lenyomását a megfelelő bit 1-be váltása jelzi, mert a leolvasó rutin negálja a leolvasott értéket.

A memória kiolvasásakor többször le kell olvasni a billentyűzetet, mert a megszakításrutin nemcsak beolvassa, hanem törli is az adatokat, valamint az önmagukban az editor számára értéktelen (CTRL, SHIFT stb.) billentyűket nem tudjuk így leolvasni.

A sorszám kiküldésekor vigyázni kell, mert a 7. bit a REM2, a 6. pedig a REM1 vezérlője: ha a bit 1, akkor a kimenet aktív. (A vezérlő jelet azonban a megszakításrutin is kiküldi, tehát ha nincs letiltva a megszakítás, akkor nem érdemes itt próbálkozni.)

Az ENTERPRISE számítógép legfelső képernyősora egy ún. STATUS sor, amely a program pillanatnyi állapotára vonatkozó adatokat tartalmazza. BASIC-ből ezt a sort csak ki, ill.

A mátrix felépítése:

Sor-	Memóriacím				BITEK			
szám:	(255. lap)	0.	1.	2.	3.	4.	5.	6.	7.
0.	15638	N		B	C	V	X	Z	bal SHIFT
1.	15639	H	LOCK	G	D	F	S	A	CTRL
2.	15640	U	Q	Y	R	T	E	W	TAB
3.	15641	7	1	6	4	5	3	2	ESC
4.	15642	F4	F8	F3	F6	F5	F7	F2	F1
5.	15643	8	9	9	–	0		ERASE	
6.	15644	5		K	;	L	:		
7.	15645	STOP		–		HOLD		ENT	ALT
8.	15646	M	DEL	,	/	.	jobbS	SPC	INS
9.	15647	I		O	@	P			

A sorszámot kell kiküldeni a 181-es portra.

```
  10 PROGRAM DEMO_PRGM
 100 ALLOCATE 100
 110 CODE ST=HEX$("0E,81,F3,CB,03,CB,03,CB,03,ED,59,CB,0B,CB,0B,
CB,0B,1C,3E,03,D3,B5,DB,B5,E6,80,06,0B,10,FE,0,0,20,E1,FB,C9")
 190 DEF STATUS(A$)
 200   FOR I=1 TO LEN(A$)
 210     SPOKE 255,16055+I,ORD(A$(I))
 220   NEXT I
 230 END DEF
1000 REM Foprogram
1010 LET X$="     Szabo Lajos Kecskemet      ":CALL STATUS(X$)
1020 OUT 191,2
1030 REM Grafikus lap letrehozasa
1040 SET VIDEO Y 24
1045 SET VIDEO MODE 1
1050 SET VIDEO COLOR 0
1055 OPEN #10:"video:"
1060 SET #10:PALETTE BLACK,WHITE
1065 SET #10:INK 1
1070 DISPLAY #10:FROM 1 TO 24 AT 1
1200 PLOT #10:0,0;1279,0;1279,863;0,863;0,0;
1210 PRINT #10,AT 10,10:"A teljes kep grafikus"
1220 WAIT 3
1300 LET A$="     STOP : ESCAPE billentyu      "
1310 CALL STATUS(A$)
1320 LET A=USR(ST,0)
1330 LET X$="          ":CALL STATUS(X$):! 6. kar. torlese
1340 DISPLAY TEXT
1350 END
```

A gépi kódú rutin assembler listája:

```
START  0E,81   LD C,81       ; A BORDER portja
       F3      DI
LOOP   CB,03   RLC E         ; A léptetésekre
       CB,03   RLC E         ; A színek
       CB,03   RLC E         ; sorrendje miatt
       ED,59   OUT (C),E     ; van szükség
       CB,0B   RRC E
       CB,0B   RRC E
       1C      INC E         ; köv. szín
       3E,03   LD A,3        ; 3. bill. sor
       D3,B5   OUT (B5),A
       DB,B5   IN A,(B5)     ; bill. leolv.
       E6,80   AND 80        ; csak az ESC.
       06,0B   LD B,0B       ; bill.
DEL    10,FE   DJNZ DEL      ; időzítés
       00      NOP
       00      NOP
       20,E1   JR NZ, LOOP   ; Ha nem ESC
       FB      EI            ; akkor ugrás
       C9      RET
```

bekapcsolni tudjuk. Egy kis munkával azonban mi is tudunk ide írni. A sor tartalmát a 255. lap 16056–16096 memória tartalmazza. Ha ide megfelelő értéket POKE-olunk be, átírhatjuk a sor tartalmát.

Azonban: az editorba való visszatérés átírja a sort a 6. karakter kivételével, valamint a megszakításrutin az első 5 karaktert használja, BASIC programból tehát az első 5 karaktert ne használjuk.

A SET BORDER n utasítás csak a 27-es rendszerváltozót (BORD VID) változtatja meg, a keretszínt a megszakításrutin állítja át a soron következő lefutása során. A 129. portra egy új keretszínt kiküldve, az azonnal megjelenik a képernyőn, de ha a BORD VID rendszerváltozót nem írjuk át, a megszakításrutin természetesen visszaállítja az eredeti keretszínt. Így érdekes hatások hozhatók létre, gépi kódból teljesen kihasználhatjuk a lehetőséget.

A felhasználói kézikönyvben a függőleges felbontás maximumát 720 pontban határozzák meg. Ez így van, ha a grafikus képernyőt GRAPHICS utasítással hozzuk létre. Ha azonban OPEN-nel megnyitunk egy grafikus csatornát és a lap mérete függőlegesen 24 karakterre van állítva (SET VIDEO Y 24) valamint az ablakot is 24 karakterre definiáljuk (DISPLAY csat.: FROM 1 TO 24 AT 1) akkor a függőleges felbontás 864 pont.

A programban példa található a nagyobb függőleges felbontásra, a STATUS sor írására, a BORDER színezésére és gépi kódból történő billentyűzet leolvasására. Egyébként a programból látható, hogy milyen strukturált programokat lehet előállítani a gép BASIC-jével. (Akik még nem találkoztak a géppel, azoknak elmondom, hogy a lista tördelését a gép automatikusan végzi el, a lista azonos a képernyőn megjelenővel.)

A 191. port 1. bitjével megváltoztathatjuk a billentyűhangot.

Próbáljuk ki: OUT 191,2

Szabó Lajos

NÉHÁNY KIEGÉSZÍTÉS A FELHASZNÁLÓI KÉZIKÖNYVHÖZ

Nem célom a gépkönyv részletes bírálata, ahhoz ugyanis a lap terjedelme túlságosan kicsiny. Sajnos, amellett, hogy rendkívül kevés információt ad a gyakorlottabb felhasználónak, igen durva sajtóhibákkal keseríti meg a gyanútlan kezdő életét is (már persze a gépkönyv).

I. Az INPUT utasításban, ha a kérdőjel helyett szöveget akarunk megjeleníteni, természetesen a PROMPT kulcsszót kell alkalmaznunk, nem pedig a könyvben többször is (és mindannyiszor hibásan) szereplő PROMT-ot. Hibás tehát a 18. oldalon kezdődő első példaprogram hat helyen, a második két helyen, a 26. oldal programja két helyen, a 27. oldal szövege három helyen...

II. A POS(x$,y$) és a POS (x$,y$,n) belső függvények a felhozott példákkal együtt hibásak. Helyesen a POS(x$, y$) az y$ szövegrészlet első előfordulását adja meg az x– szövegben, éppen fordítva, mint a leírásban szerepel. Ugyanez igaz a POS(x$, y$, n)-re is. A példák helyesen:
POS(„LONDON", „ON")=2, és
POS(„LONDON", „ON", 4)=5

III. Kimaradt két (bár nem túl jelentős) belső változó a felsorolásból:
VERNUM értéke a verziószám, azaz 2.1
VERS „Enterprise IS–BASIC version 2.1 c 1985 Intelligent Software Ltd"
Van még a könyvben sajtóhiba bőven, de azok nem ilyen súlyosak, így a fent említettek miatt nem sorolom fel őket. Végezetül az általam ismert speciális karakterkódokat adom meg, amelyek nagyon jól használhatók programból a képernyőtartalom formázására és mozgatására:
24 a bal margót a kurzor oszlopába állítja
25 töröl a sor végéig (mint a Shift-Delete)
26 törli a képernyőt

160 Delete
161 Shift+Delete
162 Ctrl+Delete
163 Alt+Delete
164 Erase
165 Shift+Erase
166 Ctrl+Erase
167 Alt+Erase
168 Insert
169 Shift+Insert
170 Ctrl+Insert
171 Alt+Insert

176-tól 191-ig találhatók a kurzormozgató kódok:

176 föl	**177 Sh+föl**
178 Ct+föl	**179 Al+föl**
180 le	**181 Sh+le**
182 Ct+le	**183 Al+te**
184 balra	**185 Sh+balra**
186 Ct+balra	**187 Al+balra**
188 jobbra	**189 Sh+jobbra**
190 Ct+jobbra	**191 Al+jobbra**

240 Ctrl+F1	**241 Ctrl+F2**
242 Ctrl+F3	**243 Ctrl+F4**
244 Ctrl+F5	**245 Ctrl+F6**
246 Ctrl+F7	**247 Ctrl+F8**
248 Alt+F1	**249 Alt+F2**
250 Alt+F3	**251 Alt+F4**
252 Alt+F5	**253 Alt+F6**
254 Alt+F7	**255 Alt+F8**

Mészáros Gyula
1029 Bp Zsíroshegyi út 110.

BRUTTÓSÍTÁS ISMÉT

Előző számunkban közöltünk egy Bruttósító programot. Amikor a kéziratot és a programlistát nyomdába adtuk, még meglehetős bizonytalanság uralkodott a személyi jövedelemadó ügyében, így több sajtóközleményben is hibásan jelentek meg az adatok. Márpedig a program e közlemények alapján készült. Nem véletlen tehát, hogy hibák kerültek bele. Ezúttal nem programozási, hanem számszaki jellegűek. Tévedéseinkre Vígh György hívta fel figyelmünket – segítségét ezúton is köszönjük. Olvasóinktól pedig elnézést kérünk. A közölt program legnagyobb hibája, hogy nem a jelenlegi nettó keresetből indul ki a számolásnál. Másrészt pedig feltételes utasításba a jövedelemadó sávhatárok kerültek, holott ehelyett a jövedelemadó-sávokhoz tartozó nettó kereset-határokat kellett volna figyelembe venni! (Aki nem érti ezt a mondatot, ne csüggedjen. Más dolog programot írni és más dolog az adózási számügyekben eligazodni. Mi sem értjük még pontosan, de abban végre biztosak vagyunk, hogy ez a program valóban azt számolja ki, amit kell!) Ezeket a hivatalosan megállapított értékeket egyébként sok lap leközölte, többek közt a 42-es Magyar Közlöny, s ezek az értékek vannak most már a programban.

```
10  rem *** bruttositas ***
20  print chr$(147)
30  print:print:print
40  print"mi a feladat ?"
50  print"1 - szamolas havi atlagbol"
60  print"2 - szamolas eves osszegbol"
70  print"3 - szamolas havi keresetekbol"
80  print:print
90  input q
100 ifq=1thengoto240
110 ifq=2thengoto180
120 ifq=3thengoto140
130 goto20
140 y=0:n=0:fori=1to12:printi;
150 input". havi brutto kereset";q:y=y+q
160 input"nyugdijjarulek";q:n=n+q
170 nexti:x=y-n:goto420
180 rem *** szamolas eves osszegbol ***
190 print chr$(147)
200 input"jelenlegi eves brutto kereset";y
210 input"jelenlegi eves nyugdijjarulek";n
220 x=y-n
230 goto420
240 rem *** szamolas havi atlagbol ***
250 printchr$(147)
260 input"jelenlegi havi brutto kereset";y
270 rem *** jelenlegi nyugdij-kulcs ***
280 n=0.03
290 ify>2100 thenn=0.04
300 ify>2600 thenn=0.05
310 ify>3300 thenn=0.06
320 ify>4300 thenn=0.07
330 ify>5300 thenn=0.08
340 ify>6300 thenn=0.09
350 ify>7300 thenn=0.11
360 ify>8300 thenn=0.12
370 ify>10300thenn=0.13
380 ify>12300thenn=0.14
390 ify>14300thenn=0.15
400 x=y-int(n*y+.5):y=y*12:print"havi netto:";x
410 x=x*12
420 rem **** adokulcs ****
430 ifx>0      thena=0   :c=0      :sa=0
440 ifx>54000 thena=0.2 :c=0      :sa=48000
450 ifx>69400 thena=0.25:c=4400   :sa=70000
460 ifx>82400 thena=0.3 :c=9400   :sa=90000
470 ifx>100400thena=0.35:c=18400  :sa=120000
480 ifx>116900thena=0.39:c=28900  :sa=150000
490 ifx>132200thena=0.44:c=40600  :sa=180000
500 ifx>159800thena=0.48:c=67000  :sa=240000
510 ifx>210200thena=0.52:c=124600:sa=360000
520 ifx>301400thena=0.56:c=249400:sa=600000
530 ifx>369400thena=0.6 :c=361400:sa=800000
540 rem *** bruttositas ***
550 b=(x+c-(sa+12000)*a)/(0.9-a)
560 print"uj eves brutto kereset:";b
570 print"uj havi brutto kereset:";b/12
580 goto30
```

Szeretnénk megnyugtatni tisztelt TVC-tulajdonos olvasóinkat, hogy a hiba nem bennünk van. Szeptemberi számunkban közöltük ugyanis a TVC DISASSEMBLER programot, amelyet sokan megpróbáltak beírni, de már az első soroknál elakadtak. A listákba ugyanis olyan kapitális méretű baromságok kerültek, amiket még az 5 évesek sem esznek meg. (Például: RETUR, vagy ON NY GOTO 3850340.) Nos a dolog – márमint, hogy jelenhetett meg ilyen hibás lista – annyira érdekes, s ráadásul technikai vonatkozású, hogy megér egy pár sort.

A lista amelyet nyomdába adtunk – természetesen jó. (Reméljük, hogy hibátlan.) A figyelmes szemlélő észreveheti, hogy mindkét lista középső része hiányzik. Mintha valaki hosszában kettévágta volna a listákat, s kiemelt volna belőle egy darabot. Nos, a hiba elektronikus „jellegű". A lapot előkészítő Zrínyi Nyomdában a fekete-fehér képek, rajzok, s így a programlisták filmjét egy elektronikus Scanner nevű berendezéssel készítik. A Scanner, úgy tűnik, időnként hibázik. (Mint megtudtuk, nem ez volt az első eset, csak nekünk még nem volt hozzá szerencsénk.) A hiba eredménye, hogy a filmkészítésnél „ugrik" a gép, és így a fotózandó kép egy sávja kimarad. Az eredmény meggyőző. Nyilván egy fotónál észrevehető a dolog, egy programlistánál ránézésre nem látszik. (Mi is csak a megjelenés után egy héttel érkező első reklamáló levél hatására döbbentünk rá, hogy valami nem stimmel.) Úgy látszik, a berendezés öregszik, mert ha jól megnézik, ugyanebben a lapszámban a 20. oldalon az Enterprise-ról készült rajzban is van egy kis „folytonossági hiány". Ennek oka ugyanaz, mint a programlista hiányosságáé.

Ha nem computeres lap lennénk, bízvást mondhatnánk, hogy „így bízzon az ember a modern technikában". Ezzel azonban magunk alatt vágnánk a fát, így hát csak azt mondjuk: íme, a magyar technológiai színvonal.

Gondolkodtunk, hogy a program iránt érdeklődő olvasóink kielégítésére csak a program közepét közöljük újra, de azután fölrémlettek előttünk a sötét utcasarkon ránk váró tévécések kezében lévő husángok, s ezért az egész program újraközlése mellett döntöttünk. Elnézést pedig ezúttal nem a magunk nevében kérünk, hiszen nem mi hibáztunk. SCANNER SORRY!

A szerkesztő azért van,

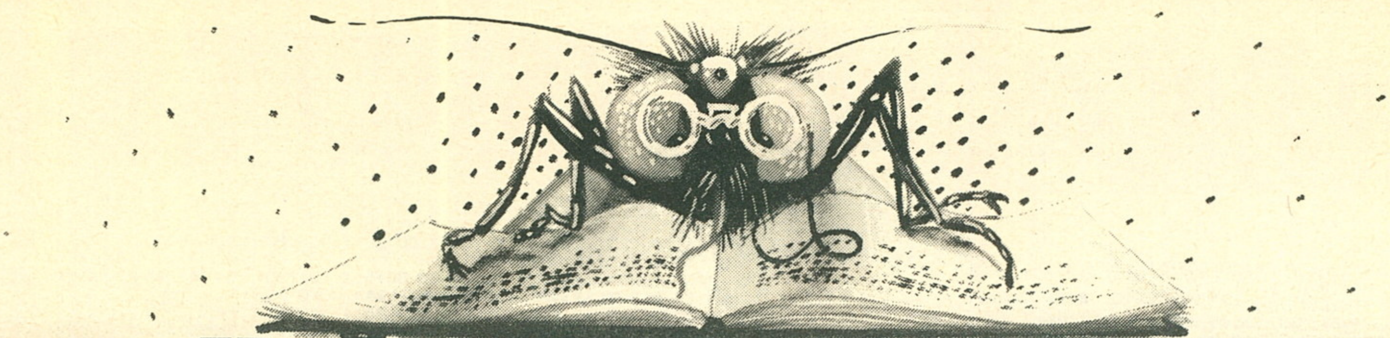
hogy a lap olyan legyen,

amilyenek az olvasói!

ÖRDÖGI

```
3010 GRAPHICS 4
3020 SET PALETTE 81,0,65,20;BORDER 17
3030 TINTA=1:PAPIR=0
3040 SET INK TINTA;PAPER 3
3050 PRINT AT 1,3:STRING$(28,126)
3060 PRINT AT 2,3:"* TVC Z-80 DISASM (C) Sz.J *"
3070 PRINT AT 3,3:STRING$(28,126)
3080 SET PAPER PAPIR
3090 PLOT 32,662;991,662,32,658;991,658
3100 PC=0:PAGING=112:NY=1:MOD=1
3110 GOSUB 6150:GOSUB 6170
3120 GOSUB 6190:GOSUB 6230
3130 GOSUB 6050
3140 DIM OP$(39)*8,V$*45,Q$*32,Q1$*32
3150 LEPES$="DANP WBBTT":M=1
3160 GET X$
3170 FOR I=1 TO LEN(LEPES$)
3180 IF X$=LEPES$(I) THEN M=I:GOTO 3200
3190 NEXT I:M=MOD
3200 ON M GOSUB 3590,3570,3500,3540,3510,3700,3760,3780,3860,3880
3210 GOTO 3160
3500 NY=((NY-1) XOR 1)+1:GOTO 6200
3510 ON NY GOTO 3530,3520
3520 LPRINT
3530 PRINT AT 9,1:CHR$(25):RETURN
3540 PRINT AT 7,19:"";:INPUT PROMPT "Paging=":Q1$:Q1$=Q1$(:2)
3550 GOSUB 6110:PAGING=Q1:GOSUB 6090:GOSUB 6180
3560 PRINT AT 7,19:STRING$(12,32):RETURN
3570 PRINT AT 7,19:"";:INPUT PROMPT "Address=":Q$:Q$=Q$(:4)
3580 GOSUB 6130:PC=Q:GOSUB 6160:GOTO 3560
3590 MOD=M:GOSUB 6230:P_C=PC:GOSUB 5000:GOSUB 3530:GOSUB 6330
3600 PC=PC+L*INT(A/140)+INT((A-INT(A/140)*140)/35)
3610 FOR I=P_C TO PC
3620 CIM=I:GOSUB 6070:W1=ADAT:GOSUB 6010:V$(7+2*(I-P_C):8+2*(I-P_C))=W1$
3630 GOSUB 6290:V$(17+I-P_C)=CHR$(ADAT)
3640 NEXT I
3650 ON NY GOTO 3670,3660
3660 LPRINT V$
3670 PRINT V$(:5)&V$(7:15)&V$(27:42)
3680 PC=PC+1:GOSUB 6160
3690 RETURN
3700 MOD=M:GOSUB 6230
3710 GOSUB 6340
3720 W=PC:GOSUB 6030:V$(:4)=W$
3730 CIM=PC:GOSUB 6070:W=ADAT
3740 CIM=PC+1:GOSUB 6070:W=W+256*ADAT:GOSUB 6030:V$(32:35)=W$
3750 V$(27:28)="DW":P_C=PC:PC=PC+1:GOSUB 3530:GOSUB 6330:GOTO 3610
3760 MOD=M+1:GOSUB 6230
3770 GOSUB 6310:X=INT(X):IF X<1 OR X>12 THEN 3770
3780 PRINT AT 4,18:"";:PRINT USING "##":X:GOSUB 6340:FOR I=1 TO X
3790 CIM=PC+I-1:GOSUB 6070:W1=ADAT:GOSUB 6010
3800 V$(6+2*(I-1):7+2*(I-1))=W1$
3810 NEXT I
3820 W=PC:GOSUB 6030:V$(:4)=W$:PC=PC+X:GOSUB 6160:GOSUB 3530
3830 GOSUB 6330:ON NY GOTO 3850,3840
3840 LPRINT V$
3850 PRINT V$(:30):RETURN
3860 MOD=M+1:GOSUB 6230
3870 GOSUB 6310:X=INT(X):IF X<1 OR X>24 THEN 3870
3880 GOSUB 6340:PRINT AT 4,18:"";:PRINT USING "##":X
3890 FOR I=1 TO X
3900 CIM=PC+I-1:GOSUB 6070:GOSUB 6290:V$(5+I)=CHR$(ADAT)
3910 NEXT I:GOSUB 3530:W=PC:GOSUB 6030:V$(:4)=W$:PC=PC+X
3920 ON NY GOTO 3940,3930
3930 LPRINT V$
3940 GOSUB 6330:PRINT V$(:30)
3950 GOTO 6160
5000 RESTORE 8000:FOR I=1 TO 39:READ OP$(I):NEXT I
5010 D=0:L=0:A=0:GOSUB 6340
5020 CIM=PC:GOSUB 6070:X=ADAT:W=PC:GOSUB 6030:V$(:4)=W$
5030 CIM=PC+L+1:GOSUB 6070:W1=ADAT:GOSUB 6010:OP$(21)=W1$
5040 OP$(17)(2:3)=OP$(21)
5050 CIM=PC+1:GOSUB 6070:W=PC+2+ADAT+256*(ADAT>127):GOSUB 6030
5060 OP$(19)=W$:CIM=PC+2:GOSUB 6070:W=256*ADAT
5070 CIM=PC+1:GOSUB 6070:W=W+ADAT:GOSUB 6030:OP$(20)=W$
5080 OP$(18)(2:5)=W$:IF D=1 THEN 5330
5090 ON X/64+1 GOTO  5100,5200,5110,5120
5100 RESTORE 8100:GOTO 5130
5110 RESTORE 8200:GOTO 5130
5120 RESTORE 8300
5130 FOR I=0 TO X AND 63:READ A,B:NEXT I
5140 IF A+B=0 THEN 5230
5145 RESTORE 8400:IF (X AND 7)=7 THEN W1=(X AND 56):GOSUB 6010:OP$(17)=W1$
5150 C=A-35*INT(A/35):FOR I=1 TO C:READ C$:NEXT I
5160 C$=C$&STRING$(5-LEN(C$),32):V$(27:31)=C$
5170 C=B-40*INT(B/40):IF C=0 THEN RETURN
5180 X$=OP$(C):C=INT(B/40):IF C<>0 THEN X$=X$&","&OP$(C)
5190 V$(32:)=X$&STRING$(12-LEN(X$),32):RETURN
5200 IF X=118 THEN V$(27:30)="HALT":RETURN
5210 V$(27:28)="LD":X$=OP$((X AND 56)/8+1)&","&OP$((X AND 7)+1)
5220 A=140:GOTO 5190
5230 IF X=237 THEN 5310
5240 IF X=203 THEN 5470
5250 IF X=221 THEN OP$(7)="(IX+00)":OP$(11)="IX":OP$(38)="(IX)"
5260 IF X=253 THEN OP$(7)="(IY+00)":OP$(11)="IY":OP$(38)="(IY)"
5270 CIM=PC+2:GOSUB 6070:OP$(7)(4)=CHR$(43-2*(ADAT>127))
5280 W1=(ADAT>127)*256+ADAT:W1=ABS(W1):GOSUB 6010
5290 OP$(7)(5:6)=W1$:L=1:PC=PC+1:CIM=PC:GOSUB 6070
5300 X=ADAT:GOTO 5030
5310 PC=PC+1:CIM=PC:GOSUB 6070:X=ADAT:D=1:L=0
5320 OP$(11)="HL":OP$(7)="(HL)":GOTO 5030
5330 D=0:IF ((X XOR 66) AND 207)<>0 THEN 5350
5340 V$(27:29)="SBC":B=371+5*(X AND 48)/2:GOTO 5170
5350 IF ((X XOR 74) AND 207)<>0 THEN 5370
5360 V$(27:29)="ADC":B=371+5*(X AND 48)/2:GOTO 5170
5370 IF ((X XOR 64) AND 199)<>0 THEN 5390
5380 V$(27:28)="IN":B=681+(X AND 56)/8:OP$(17)="(C)":GOTO 5170
5390 IF ((X XOR 65) AND 199)<>0 THEN 5410
5400 V$(27:29)="OUT":B=57+5*(X AND 56):OP$(17)="(C)":GOTO 5170
5410 IF ((X XOR 67) AND 207)<>0 THEN 5430
5420 V$(27:28)="LD":B=378+5*(X AND 48)/2:PC=PC+2:GOTO 5170
5430 IF ((X XOR 75) AND 207)<>0 THEN 5450
5440 V$(27:28)="LD":B=729+(X AND 48)/16:PC=PC+2:GOTO 5170
5450 RESTORE 8500
5460 READ C$,C:IF C*X=C*C THEN V$(27:)=C$:RETURN:ELSE 5460
5470 PC=PC+1+L:CIM=PC:GOSUB 6070:X=ADAT
5480 ON X/64+1 GOTO 5490,5500,5510,5520
5490 RESTORE 8600:A=1+(X AND 56)/8:B=1+(X AND 7):GOTO 5150
5500 V$(27:29)="BIT":GOTO 5530
5510 V$(27:29)="RES":GOTO 5530
5520 V$(27:29)="SET"
5530 B=(X AND 56)/8+30+((X AND 7)+1)*40:GOTO 5170
6000 W2$=CHR$(W2+48-7*(W2>9)):RETURN
6010 W2=INT(W1/16):GOSUB 6000:W1$=W2$:W2=W1-16*W2:GOSUB 6000
6020 W1$=W1$&W2$:RETURN
6030 W1=INT(W/256):GOSUB 6010:W$=W1$:W1=W-256*W1:GOSUB 6010
6040 W$=W$&W1$:RETURN
6050 RESTORE 8700:KOD$=""
6060 FOR I=1 TO 14:READ X:KOD$=KOD$&CHR$(X):NEXT I:RETURN
6070 CIM=CIM+(CIM>32767)*65536
6080 ADAT=USR(2+VARPTR(KOD$),CIM):RETURN
6090 POKE VARPTR(KOD$)+4,PAGING:RETURN
6100 Q2=ORD(Q2$)-48+7*(Q2$>"9"):RETURN
6110 Q2$=Q1$(:1):GOSUB 6100:Q1=Q2:Q2$=Q1$(2:):GOSUB 6100
6120 Q1=16*Q1+Q2:RETURN
6130 Q1$=Q$(:2):GOSUB 6110:Q=Q1:Q1$=Q$(3:):GOSUB 6110
6140 Q=Q*256+Q1:RETURN
6150 SET INK 2:PRINT AT 6,5:"A";:SET INK TINTA:PRINT "ddress:"
6160 W=PC:GOSUB 6030:PRINT AT 6,13:W$:RETURN
6170 SET INK 2:PRINT AT 6,20:"P";:SET INK TINTA:PRINT "aging:"
6180 W1=PAGING:GOSUB 6010:PRINT AT 6,27:W1$:RETURN
6190 SET INK 2:PRINT AT 7,5:"N";:SET INK TINTA:PRINT "yomtató:";
6200 PRINT AT 7,14:"";:ON NY GOTO 6210,6220
6210 PRINT "ki":RETURN
6220 PRINT "be":RETURN
6230 PRINT AT 4,10:">> ";
6240 ON MOD GOTO 6250,6260,6260,6260,6260,6270,6280,6280,6320,6320
6250 PRINT "Disasm <<":RETURN
6260 FETURN
6270 PRINT " Word  <<":RETURN
6280 PRINT "Byte   <<":RETURN
6290 IF ADAT<32 OR ADAT>159 THEN ADAT=46
6300 RETURN
6310 PRINT AT 7,19:"";:INPUT PROMPT "Száma=":X:GOTO 3560
6320 PRINT "Text   <<":RETURN
6330 PRINT AT 23,2:"";:RETURN
6340 V$=STRING$(45,32):RETURN
8000 DATA B,C,D,E,H,L,(HL),A,BC,DE,HL,SP,AF,(BC),(DE),(SP),(00),(0000)
8010 DATA 0000,0000,00,NZ,Z,NC,C,PO,PE,P,M,0,1,2,3,4,5,6,7,(HL),HL
8100 DATA 11,0,71,809,1,334,24,9,24,1,23,1,36,841,13,0
8110 DATA 33,533,22,371,1,568,23,9,24,2,23,2,36,842,15,0
8120 DATA 66,19,71,810,1,335,24,10,24,3,23,3,36,843,12,0
8130 DATA 65,19,22,411,1,608,23,10,24,4,23,4,36,844,14,0
8140 DATA 65,782,71,811,71,458,24,11,24,5,23,5,36,845,9,0
8150 DATA 65,783,22,451,71,731,23,11,24,6,23,6,36,846,8,0
8160 DATA 65,784,71,812,71,338,24,12,164,7,163,7,176,847,16,0
8170 DATA 65,785,22,491,71,728,23,12,24,8,23,8,36,848,7,0
8200 DATA 22,48,22,88,22,128,22,168,22,208,22,248,162,288,22,328
8210 DATA 27,48,27,88,27,128,27,168,27,208,27,248,167,288,27,328
8220 DATA 20,1,20,2,20,3,20,4,20,5,20,6,160,7,20,8
8230 DATA 26,48,26,88,26,128,26,168,26,208,26,248,166,288,26,328
8240 DATA 17,1,17,2,17,3,17,4,17,5,17,6,157,7,17,8
8250 DATA 21,1,21,2,21,3,21,4,21,5,21,6,161,7,21,8
8260 DATA 19,1,19,2,19,3,19,4,19,5,19,6,159,7,19,8
8270 DATA 18,1,18,2,18,3,18,4,18,5,18,6,158,7,18,8
8300 DATA 28,22,34,9,99,822,99,20,95,822,10,9,57,848,32,17
8310 DATA 28,23,28,0,99,823,0,0,95,823,95,20,62,848,32,17
8320 DATA 28,24,34,10,99,824,37,337,95,824,10,10,55,21,32,17
8330 DATA 28,25,6,0,99,825,38,688,95,825,0,0,61,848,32,17
8340 DATA 28,26,34,11,99,826,33,456,95,826,10,11,52,21,32.17
8350 DATA 28,27,29,38,99,827,33,1570,95,827,0,0,56,21,32,17
8360 DATA 28,28,34,13,99,828,4,0,95,828,10,13,54,21,32,17
8370 DATA 28,29,1,452,99,829,5,0,95,829,0,0,53,21,32,17
8400 DATA LD,OUT,IN,DI,EI,EXX,CCF,CPL,DAA,PUSH,NOP,RLA,RLCA
8410 DATA RRA,RRCA,SCF,AND,CP,OR,SUB,XOR,ADD,DEC,INC,CALL
8420 DATA SBC,ADC,RET,JP,JR,DJNZ,RST,EX,POP
8500 DATA NEG,68,RETN,69,IM0,70,"LD   I,A",71,RETI,77,"LD   R,A"
8510 DATA 79,IM1,86,"LD   A,I",87,IM2,94,"LD   A,R",95,RRD
8520 DATA 103,RLD,111,LDI,160,CPI,161,INI,162,OUTI
8530 DATA 163,LDD,168,CPD,169,IND,170,OUTD,171,LDIR,176
8540 DATA CPIR,177,INIR,178,OTIR,179,LDDR,184,CPIR,185
8550 DATA INDR,186,OTDR,187,"Not used",0
8600 DATA RLC,RRC,RL,RR,SLA,SRA,SLL,SRL
8700 DATA 243,62,112,211,2,110,38,0,62,112,211,2,251,201
```

ötlet



KÖNYVMOLY

Rovatunk ezúttal rendhagyó módon jelentkezik – ez bizonyára rögtön látszik, hiszen korábban nem közöltünk programlistákat vagy grafikákat. Olvastunk ugyanis egy könyvet, amely annyira felkeltette érdeklődésünket, hogy azonnal ki is próbáltuk a leírt algoritmusokat, és megnéztük, melyikből mi is „jön ki". Először azonban tekintsük át szokás szerint a hónap könyvújdonságait:

Grocmann–Eichler: A 8086/88-as mikroprocesszor – Data Becker – Novotrade, 357. o., 349 Ft.
(A könyv az Intel 8086-os, illetve iAPX86-os mikroprocesszorcsalád tagjainak felhasználásához, alkalmazásához nyújt segítséget.

Bognár Júlia: dBASE III PLUS – Számalk, 256 o., 156 Ft.
(Az előismeretek nélkül is jól használható kötet a széles körben elterjedt dBASE III adatbáziskezelő program továbbfejlesztett változatát mutatja be.)

Bodor Tibor: A Commodore 64 programozásának gyakorlata III. Számalk, 187 o., 137 Ft.
(A sorozat harmadik kötete a közvetlen elérésű – random vagy relatív – lemezállományok kezelésével foglalkozik, a lemezkezelő parancsokon kívül kitérve programtervezési, adatfeldolgozási kérdésekre is. Az állományok használatát működő mintaprogramok illusztrálják.)

IBM PC DOS I–III. – LSI ATSZ, 1005 o., 692 Ft.
(A háromkötetes, kimerítő ismereteket tartalmazó könyv első része a PC DOS használatát, a második a felépítését ismerteti, a harmadik pedig az IBM kompatíbilis mikrogépek leghatékonyabb programfejlesztő eszközének, a macro assemblernek használatát segíti.)

Liesert: PEEK-ek és POKE-ok a C 64-esen – Data Becker – Novotrade, 166 o, 120 Ft.
(A szerző a gép nulláslapjának használatához ad sok ötletes tanácsot, működésképes trükköt.)

Babán Gábor–Masa István: Gépi kódú programozás kezdőknek és haladóknak C 16 és PLUS/4 számítógépekre – Novotrade, 211 o., 129 Ft.
(A kötet első része tankönyv, melyből önállóan is elsajátítható a gépi kódú programozás. A második rész a két gépről nyújt számos hasznos információt.)

Yoshiaka Shiari–Jun-ichi Tsujii: Mesterséges intelligencia – Novotrade, 179 o., 349 Ft.
(A könyv a mesterséges intelligencia alkalmazási területeinek alapvető fogalmaival és eljárásaival foglalkozik, valamint áttekinti a gyakorlati megvalósítások alapelveit.)

Kepes János: Mikroszámitógépes grafika – Műszaki Könyvkiadó, 157., 55 Ft.
A kötet címe kissé megtévesztő. Az olvasó ennek alapján azt is várhatja, hogy grafikával foglalkozó számítástechnikusok és számítástechnikával foglalkozó grafikusok műveit mutatja be a könyv, vagy az ő műhelytitkaikba vezet be. Az alcím – **Grafikai algoritmusok** – már többet elárul: Kepes János a grafikában járatlan, de érdeklődő, kísérletező kedvű olvasóknak írta kötetét.
A könyv egy géptípushoz **– a Spectrumhoz – kötődik,** a szerző e gép grafikai lehetőségeit igyekszik kiaknázni. A kötetet gondozó Műszaki Könyvkiadó is hamarosan forgalmazni kezdi a demonstrációs programokat tartalmazó kazettát – természetesen Spectrumra.
Ez a géphez kötöttség azonban nem túl erős, nem köti meg az olvasó kezét, mert a szerző nem közöl programlistákat. „Csupán" olyan matematikai eljárásokat, algoritmusokat mutat be, amelyeknek segítségével könnyen előállíthatók geometrikus jellegű, de mégis látványos rajzok a képernyőn. Az eljárások alapelveit írja csak le, gyakran azokat is meglehetősen nagyvonalúan, felszínesen. Így azután az olvasónak is elég alkalma marad a fejtörésre: hogyan is oldja meg egy-egy rajz előállítását a saját gépén?

EGY KIS GRAFIKA

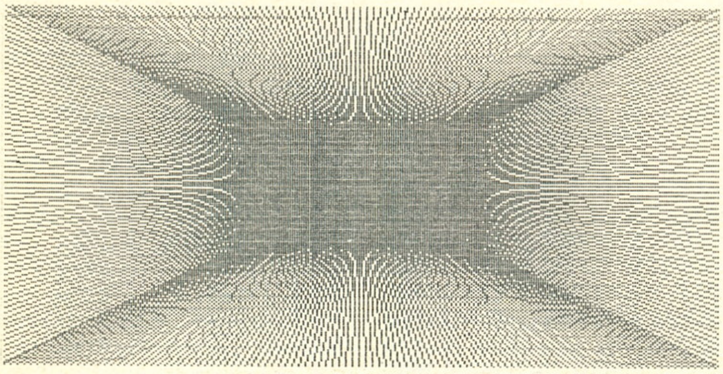
Ehhez szeretnénk segítséget nyújtani az olvasóknak, nem azért, hogy megkíméljük őket a gondolkodástól, a felesleges „okoskodástól", hanem hogy az egyes eljárások továbbgondolására biztassunk. Kis programjainkat Commodore Plus/4-esre írtuk, egyrészt, mert a Spectrum-tulajdonosok úgyis hozzájuthatnak a kész kazettához, másrészt azért, hogy megmutassuk: az eljárások más géptípusra is átírhatók.
Az első fejezet, a **bevezetés** tulajdonképpen előszónak is tekinthető, a szerző itt a grafika és a matematika összekapcsolhatóságát hangsúlyozza. A második rész az **„Előkészítés"** címet viseli, ez tartalmazza egyrészt a későbbiekhez szükséges matematikai ismereteket, a grafikai programozás alapjait, és a grafikai elemeket. A matematikai összefoglaló kissé tömörre sikeredett, követni és alaposan megérteni talán csak az tudja, aki mindezt tanulta már, és az itt leírtak csak az ismétlést szolgálják. Mindez azért hiba, mert a kötet további része gyakran ezekre az ismeretekre épít. A programozási elemek bemutatása viszont alapos és kimerítő. Innen megtudható, hogy hogyan működik a sok gépbe már előzetesen beépített egyenes- és görberajzoló, foltbetöltő és szakaszkivágó algoritmus, és hogy egy grafikai programot hogyan célszerű felépíteni. A grafikai elemek ismertetése viszont ismét csak meglehetősen felületes, általánosságokat fogalmaz meg.

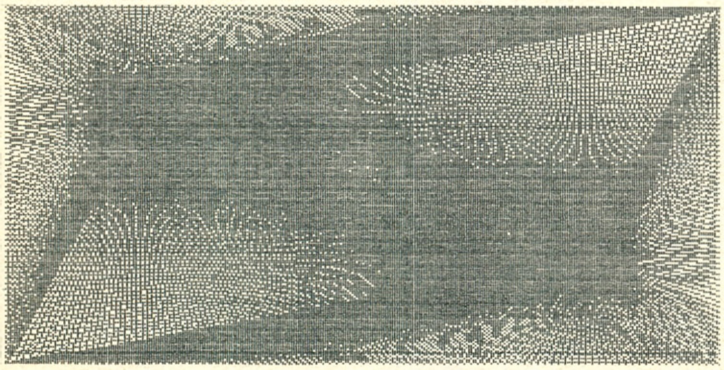
EGY KIS GRAFIKA

A könyv harmadik részében kezdődik végre az „érdemi munka": ennek alapján már bárki kedve szerint telerajzolhatja a képernyőt. A terjedelmes fejezet a **„Szabályos képek"** címet viseli, görbékből, görbeseregekből és pontokból előállítható képek algoritmusait mutatja be. Hogy már egyszerű, egyenes szakaszok seregéből is milyen látványos kép állítható elő, arra mutat példát az **1.** és a **2. program**

```
10 GRAPHIC1,1
20 FORI=0TO318STEP3
30 DRAW,I,0TO318-I,199
40 NEXTI
50 FORJ=0TO198STEP3
60 DRAW,0,JTO319,198-J
70 NEXTJ
```

1





2

```
10 GRAPHIC1,1
20 FORI=0TO318STEP3
30 DRAW,I,0TO60,60TO318-I,199TO260,140TO
I,0
40 NEXTI
50 FORJ=0TO198STEP3
60 DRAW,0,JTO60,60TO319,198-JTO260,140TO
0,J
70 NEXTJ
```

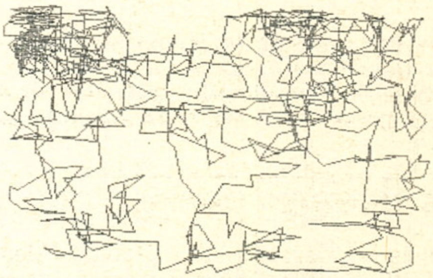
– mely a könyv „Interferenciaminták" című alfejezete alapján készült. Kiemelésre érdemes a „Képek pontokból" alfejezet is, de erre nem mutatunk be külön illusztráló programot, hiszen főként fraktálokkal és sejtautomatákkal foglalkozik, amiről a BIT-LET-ben már többször részletesen szóltunk.

A 4. fejezet a kötet legérdekesebb része: **„A véletlen képei"**. A legegyszerűbb, egyenes vonaldarabokból álló „firkálástól" kezdve eljut olyan rajzok létrehozásáig, amelyek labirintusra, talajrétegek egymásra halmozódására, tintafoltokra, vagy szövetmintára emlékeztetnek. Úgy érezzük, hogy információtartalma és sokoldalú illusztrációs felhasználhatósága miatt érdemes ezekkel az eljárásokkal részletesebben is foglalkoznunk. Azt, hogy a képernyőt hogyan lehet véletlenszerűen összefirkálni, nyilván mindenki tudja, aki megírta első önálló rajzolóprogramocskáit: sorra választunk pontokat a képernyőn, és mindegyiket összekötjük az előzővel. Ez azután továbbfejleszthető úgy, hogy a rajz egy légy vagy egy szúnyog mozgását idézze – ezt mutatja a **3.** és a **4. program.**

```
10 GRAPHIC1,1
20 X=160:Y=10
30 DRAW,X,Y
40 A=2*π*RND(1)
50 LOCATEX,Y
60 V=30*RND(1)*COS(A):W=30*RND(1)*SIN(A)
70 IFX+V<0ORX+V>256ORY+W<0ORY+W>200THENG
OTO40
80 X=X+V:Y=Y+W:DRAWTOX,Y
90 GOTO40
```

3





4

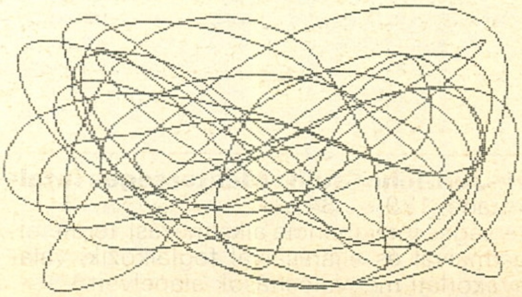
```
10 GRAPHIC1,1
20 X=160:Y=10
30 DRAW,X,Y
40 A=2*π*RND(1)
50 LOCATEX,Y
60 V=15*COS(A)*RND(1):W=15*SIN(A)*RND(1)
70 IFX+V>0ANDX+V<256ANDY+W>0ANDY+W<200TH
ENX=X+V:Y=Y+W:DRAWTOX,Y
80 GOTO40
```

Látható, hogy nem sokban különböznek, az eltérés csak annyi, hogy a légy mozgásánál az egymáshoz kapcsolódó egyenes szakaszok – a gép grafikai felbontóképességéhez mérve – megközelítőleg ugyanolyan hosszúak, a szúnyog mozgásánál pedig a hossz véletlenszerűen változik.

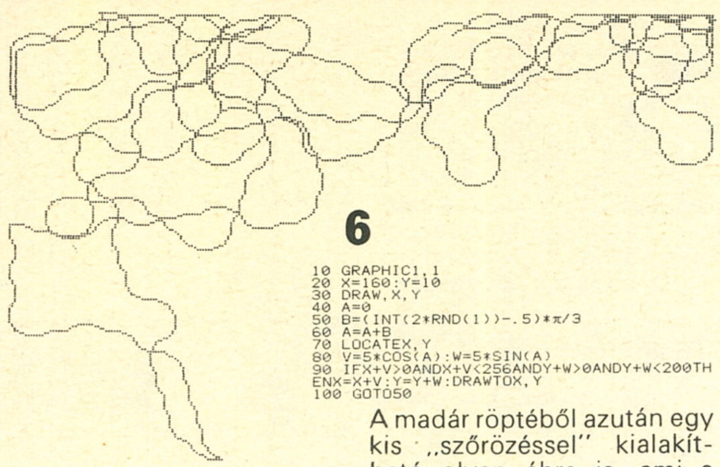
A két kis program láthatóan azonos elvre épül – de továbbfejleszthető úgy is, hogy madár, vagy pillangó repülésére emlékeztessen a rajz. Ezek röpte nem olyan cikcakkos, mint a légyé vagy a szúnyogé – így a röppályát itt pontonként rajzoljuk meg. **(5., 6. program).**

```
10 GRAPHIC1,1
20 X=160:Y=100:D=10:H=130
25 V1=10:V2=10
30 DRAW,X ,Y
45 V1=V1+SGN(RND(1)*(320-H)+H/2-X)
46 IF ABS(V1)>DTHENV1=D*SGN(V1)
47 V2=V2+SGN(RND(1)*(200-H)+H/2-Y)
48 IF ABS(V2)>DTHENV2=D*SGN(V2)
50 LOCATEX,Y
60 DRAWTOX+V1,Y+V2
65 X=X+V1:Y=Y+V2
70 GOTO 45
```

5



EGY KIS GRAFIKA



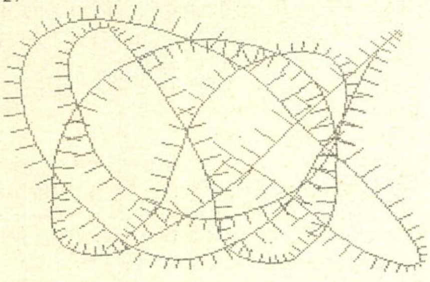
6

```
10 GRAPHIC1,1
20 X=160:Y=10
30 DRAW,X,Y
40 A=0
50 B=(INT(2*RND(1))-.5)*π/3
60 A=A+B
70 LOCATEX,Y
80 V=5*COS(A):W=5*SIN(A)
90 IFX+V>0ANDX+V<256ANDY+W>0ANDY+W<200TH
ENX=X+V:Y=Y+W:DRAWTOX,Y
100 GOTO50
```

A madár röptéből azután egy kis „szőrözéssel" kialakítható olyan ábra is, ami a majmok és az elmebetegek rajzaira emlékeztet **(7. program)** sőt, ezzel a módszerrel tekergő szalagot is rajzolhatunk. **(8. program).**

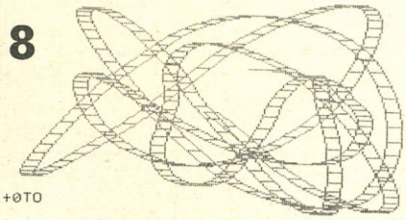
```
10 GRAPHIC1,1
20 X=160:Y=100:D=10:H=130
30 V1=10:V2=10
40 DRAW,X ,Y
50 V1=V1+SGN(RND(1)*(320-H)+H/2-X)
60 IF ABS(V1)>DTHENV1=D*SGN(V1)
70 V2=V2+SGN(RND(1)*(200-H)+H/2-Y)
80 IF ABS(V2)>DTHENV2=D*SGN(V2)
90 LOCATEX,Y
100 DRAWTOX+V1,Y+V2
110 X=X+V1:Y=Y+V2
120 DRAWTOX-V2,Y+V1:
130 GOTO 50
```

7



8

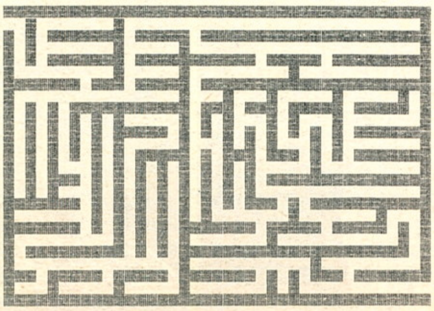
```
10 GRAPHIC1,1
20 X=160:Y=100:D=10:H=130
30 V1=10:V2=10
40 DRAW,X ,Y
50 V1=V1+SGN(RND(1)*(320-H)+H/2-X)
60 IF ABS(V1)>DTHENV1=D*SGN(V1)
70 V2=V2+SGN(RND(1)*(200-H)+H/2-Y)
80 IF ABS(V2)>DTHENV2=D*SGN(V2)
90 LOCATEX,Y
100 DRAWTOX+V1,Y+V2
110 X=X+V1:Y=Y+V2
120 DRAWTOX+8,+0TOX+8-V1,Y-V2TOX-V1,+0TO
X,Y
130 GOTO 50
```



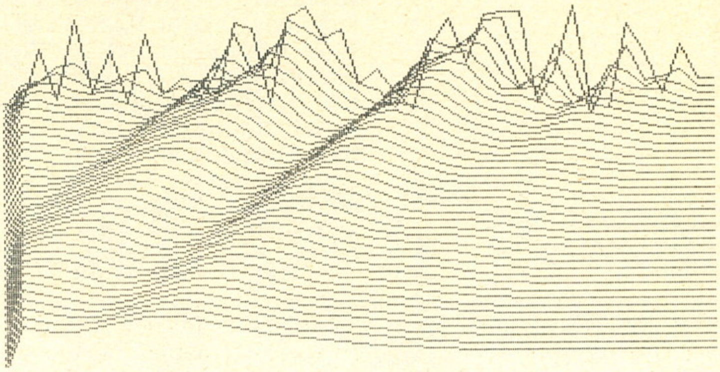
A véletlen, és a képernyőn való bolyongás problémájához tartoznak a labirintusok is: hogy tudunk számítógéppel olyan labirintust rajzolni, amely minden futtatás után más és más pályát ad, de bármely két pontja között pontosan egy út vezet – azaz nincsenek bezárt területei, körbejárható részei. Kepes János részletesen leírja az algoritmust – sőt, az már az 1985-ben megjelent, Etűdök személyi számítógépre című könyvben is szerepelt –, így csak a programlistát közöljük. **(9. program).**

```
10 PRINTCHR$(147)
20 A=3072:DIMC(10,17)
30 FORI=0TO38
40 POKE A+I,160:POKEA+960+I,160
50 NEXTI
60 FORI=80 TO 920 STEP40
70 POKE A+I,160:POKEA-2+I,160
80 NEXTI
90 Y=0
100 P=INT(11*RND(0)):Q=INT(18*RND(0))
110 IF C(P,Q)=1 THEN 100
120 R=A+82+P*80+Q*2
130 POKER,160:C(P,Q)=1:Y=Y+1
140 X=INT(4*RND(0))+1
150 ONXGOSUB180,210,240,270
160 IF Y=198 THEN 300
170 GOTO100
180 POKE R-40,160
190 IF PEEK(R-80)=160 THEN RETURN
200 R=R-80:POKER,160:P=P-1:C(P,Q)=1:Y=Y+
1:GOTO180
210 POKE R-1,160
220 IF PEEK(R-2 )=160 THEN RETURN
230 R=R-2:POKER,160:Q=Q-1:C(P,Q)=1:Y=Y+1
: GOTO210
240 POKE R+1,160
250 IF PEEK(R+2)=160 THEN RETURN
260 R=R+2:POKER,160:Q=Q+1:C(P,Q)=1:Y=Y+1
:GOTO240
270 POKE R+40,160
280 IF PEEK(R+80)=160 THEN RETURN
290 R=R+80:POKER,160:P=P+1:C(P,Q)=1:Y=Y+
1:GOTO270
300 GETKEY A$
```

9



A talajrétegek gyűrődésére emlékeztető rajz elkészítésének alapelve igen egyszerű – **10. program.** A képet



nem úgy építjük fel, ahogy a természetben létrejön, vagyis nem az alsó, sima rétegektől kiindulva „ráncosítjuk" felfelé haladva, hanem fordítva: rajzolunk egy véletlenszerűen cikcakkos vonalat a képernyő felső részén, és lefelé haladva ezt „egyengetjük" ki.

Ennyi ízelítő bizonyára elég ahhoz, hogy érzékelhető legyen, mi mindenre jó a számítógép véletlenszám-generátorának és grafikájának párhuzamos használata. Térjünk rá az 5. fejezetre, amely a **térbeli képek** megjelenítésével foglalkozik!

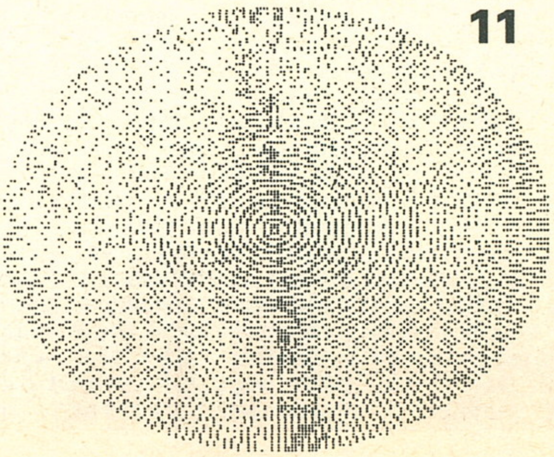
Mivel egy térbeli tárgy síkra leképezése nem kevés matematikai ismeretet igényel, a fejezet egy elméleti bevezetővel kezdődik. Ezzel ugyanaz lehet a problémánk, mint amit a második fejezet matematikai fejtegetései kapcsán említettünk: annak, aki korábban nem foglalkozott hasonló feladatokkal, nem sokat mond. Az algoritmusok viszont szép és látványos eredményt adnak – bemutatja a szerző a görbült felületek érzékeltetésének módszereit, a testek takarásának és árnyékolásának megjelenítését. Ez utóbbira mutat példát a **11. program.**

10

```
10 GRAPHIC1,1
20 DIMA(40):DIMB(40)
30 FORI=0TO40:A(I)=60*RND(1):NEXTI
40 GOSUB140
50 Q=A(0)
60 FORY=QTO200STEP4
70 FORI=1TO39:B(I)=(A(I)+A(I+1))/2+4
80 NEXTI
90 FORI=1TO39:A(I)=B(I):NEXTI
100 A(0)=A(0)+4:A(40)=A(40)+4
110 GOSUB140
120 NEXTY
130 STOP
140 FORX=1TO40
150 LOCATE(X-1)*8,A(X-1)
160 DRAWTOX*8,A(X)
170 NEXTX
180 RETURN
```

```
10 GRAPHIC1,1
20 P=160:Q=100:R=80
30 FORFI=0TO2*πSTEP.03
40 FORKSZI=0TOπSTEP.03
50 E=(SIN(FI)*COS(KSZI)+COS(FI)*COS(KSZI
)+SIN(KSZI)+2)/4
60 IFE>RND(1)THENDRAW,P+R*SIN(FI)*COS(KS
ZI),Q+R*COS(FI)*COS(KSZI)
70 NEXTKSZI:NEXTFI
```

11



Gömb képzetét kelti egy kör alakú terület megfelelő árnyalásával.
A kötet további része – 6–9. fejezet – szintén tartalmaz tetszetős illusztrációkat és jól használható eljárásokat. Ezekhez már nem közlünk programlistákat, mivel az itt bemutatott grafikák egyszerűen előállíthatóak – vagy azért, mert nagyon könnyen gépre vihetőek, vagy azért, mert a legprimitívebb rajzolóprogram is többre képes e fejezetek témái kapcsán, mint amit BASIC-ből elérhetünk.
Így a 6. fejezet, a **Szimmetrikus minták"** azt mutatja be, hogy a képernyőt hogyan tölthetjük fel azonos ábrarészletekkel, és hogy a különböző feltöltések milyen grafikai hatást eredményeznek. Ez a feltöltés azonban a lehető legegyszerűbben egy olyan rajzolóprogrammal oldható meg, amely képes a képernyő egy-egy részét átmásolni – márpedig ilyen funkcióval a legtöbb rajzprogram rendelkezik.
A 7. rész címe: **Számítógéppel rajzolt képek.** Itt azután keverednek a grafikai és a matematikai ismeretek. Az első alfejezet – Tollrajzok – ismét csak olyan kérdésekkel foglalkozik, amelyek a legegyszerűbb rajzprogrammal is megoldhatók: a pontonkénti és szakaszonkénti rajzolással.
Persze, nem érdektelen megtudni, hogy hogyan is működnek ezek a funkciók, de sok gyakorlati haszna nincs. A tónusos rajzolás már izgalmasabb, a közelítő vonalrajzolás – közelítés po-linomokkal – viszont szinte tisztán matematika, inkább a 2. fejezetbe kívánkozik.
A 8. **„Motívumokból képek"** című rész a képzőművészeti kompozíciók megalkotásának rejtelmeibe kívánja bevezetni az olvasót, hat oldalon. Foglalkozik a motívumok kombinálásával, a kerettel, a kompozíció szabályosságával és a textúrával – de persze ilyen terjedelemben igencsak felületesen, általánosságokról szólva.
Végül a 9. fejezet a mozgó képek megalkotását, **a számítógépes animációt** mutatja be. Újra egy izgalmas téma, sajnos, nagyon tömören, kilenc oldalon bemutatva – de azért elfér itt a karaktergrafikától a képernyőgörgetésen át a valódi rajzfilmek alapelemeiig sok minden. A bemutatott eljárások alapjai lehetnek akár egy játékprogram animációjának is.
Mint kiderült, a kötet nem egyenletes színvonalú. Gyakoriak a szerkesztési következetlenségek is, így gyakran logikátlanul követik egymást az elméletieskedő és a gyakorlati képalkotást célzó részek.
Hatalmas érdeme viszont a kötetnek az az ötletgazdagság és széles látókör, ahogy a szerző körbejárja ezt a szerteágazó témakört. Talán példaprogramjaink is érzékeltetik, hogy némi matematikai ismeret birtokában és kellő kíváncsisággal milyen lehetőségei vannak a mikroszámítógépek grafikájának – de mindehhez természetesen ajánljuk a könyv elolvasását is.

PROGRAM CSERE-BERE

TV-Computer játék- és felhasználói programokat cserélnék. A legapróbb programocska is érdekel.
Kiss Csaba, 9443 Petőháza, Bartók Béla u. 20.

Commodore 16-os programokat cserélnék! Keresem a Mondy On The Run és a Cuthbert In the Tombs Of Doom C 16-on futó változatát.
Murányi Szabolcs, 2400 Dunaújváros, Váci Mihály u. 11. IX/4.

Várom azoknak a **SINCLAIR SPECTRUM 128** Kbyte-os géptulajdonosoknak a jelentkezését, akik gépkezelési leírással rendelkeznek.
Tatár Árpád, 8006 Székesfehérvár, Benke F. u. 8.

Commodore 128-as gépre és 1571-es floppyra felhasználói programok – adatfeldolgozó, szövegszerkesztő stb. – cseréjéhez kapcsolatot keresek. A programok kézikönyve, leírása is érdekel.
Gömöri József, 1086 Bp., Lujza u. 1/b

Primo programokat cserélek. Várom azok jelentkezését, akik gépükhöz hardverbővítést készítettek.
Varsányi Gábor, 7500 Nagyatád, Aradi u. IX/c, IV/3.

Keresek **ZX Spectrumhoz** felhasználói, és játékprogram-leírásokat, térképeket, örökélet (POKE) kódokat. A „Sinclair Spectrum Játék és Program I–II–III megvan."
Gedő Tamás, 1165 Bp., Veres P. út. 121.

ATARI ST tulajdonosok ismeretségét keresem „intim" eszmecsere céljából.
Tóth András, 8800 Nagykanizsa, Berzsenyi u. 14/A

COMMODORE 16, PLUS/4 felhasználói és játékprogramokat cserélek! Márkatársak írjatok!
Egri Imre, 5561 Békésszentandrás, Péró u. 2.

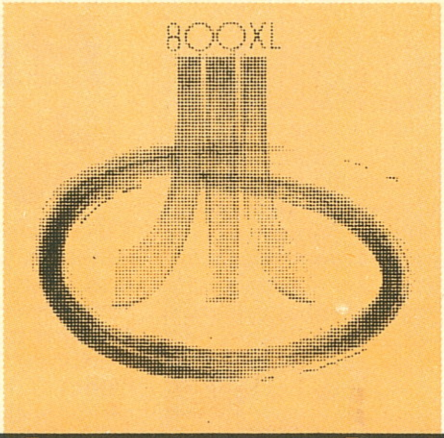
ATARI cseretársakat keresek.
Habányi Tamás, 8200 Veszprém, Egry u. 45/c

C 16, PLUS/4 gépre angol, német és orosz nyelvi oktatóprogramok eladók. Játék- és felhasználói programok cseréje.
Kálmán Albert, 3300 Eger, Rákóczi út 31. III/11. Tel.-üzenet: 143-031, 330-345 (Bp.)





ATARI KÖR



PROGRAMOZHATÓ FUNKCIÓBILLENTYŰK

Sokan – és joggal – kifogásolják, hogy az ATARI 800XL funkcióbillentyűi (HELP, START, SELECT, OPTION) BASIC-ből csak nehézkesen, PEEK függvénnyel használhatók.

Az alábbi rutin lehetővé teszi, hogy nyolc programozható billentyűt használjunk.

A=USR(37246,Fx,ADR(K$),LEN(K$))

utasítással adhatjuk meg a funkciót.

Fx helyére 1–8 közti számot kell írni:

1 – HELP
2 – START
3 – SELECT
4 – OPTION
5 – SHIFT+HELP
6 – SHIFT+START
7 – SHIFT+SELECT
8 – SHIFT+OPTION

K$ tartalmazza a kívánt szöveget; ez bármi lehet, az itt levő karakterek a továbbiakban úgy viselkednek, mintha a billentyűzetről írnánk be azokat. Ha a stringet a CHR$ (155) karakterrel zárjuk, azonnal végrehajtódó utasítást kapunk.

Pl.: K$="GRAPHIC$ 0":K$(11)=CHR$(155):A=USR (37246,3,ADR(K$),11)

Ezek után a SELECT gomb lenyomásakor – parancs módban – azonnal törlődik a képernyő és a 0. grafikai üzemmód jön létre.

Számtalan alkalmazási lehetősége van a billentyűknek, csak arra ügyeljünk, hogy a string nem lehet hosszabb, mint 32 karakter!

0 hosszúságú stringet írva a billentyűt „kikapcsolhatjuk".

Speciális lehetőség a CONTROL+HELP. Ez igazi helpfunkció! Megnyomásakor tetszőleges BASIC programsorra ugorhatunk az eredeti programfutást megszakítva!

A=USR(37321,sorszám) utasítással tölthetjük be a kívánt sorszámot, attól kezdve a CTRL+HELP gombot bármikor lenyomva a megadott sornál folytatódik a végrehajtás. Ez lényegében 'felhasználói BREAK', de lehet segédinformációk kérése, folytatás és abbahagyás vezérlője stb.

32767-nél nagyobb számot írva a billentyű hatását megszüntethetjük.

Arra kell ügyelnünk, hogy a gomb által kiváltott ugrás az esetleges TRAP sorszámot törli.

Figyelem! Ez a rutin RESET után sem törlődik, tehát elmaradhat az eddigi POKE 106,144...stb. Mondanom se kell, hogy az eddig ismertetett rutinokkal együtt használható, így most már azok is "RESET-biztosak".

Ellenőrzés:

RESTORE:A=0:FOR I=1 TO 260:READ B:A=A+B: NEXT I:? A

A képernyőn 30535-nek kell megjelennie.

Rieth József

```
10 POKE 106,144:GRAPHICS 0
20 FOR I=37246 TO 37483:READ A:POKE I,A:NEXT I
30 FOR I=37740 TO 37761:READ A:POKE I,A:NEXT I
40 A=PEEK(9):IF A<2 THEN 60
50 A=A-2:POKE 37761,76:POKE 37762,PEEK(2):POKE 37763,PEEK(3)
60 POKE 9,A+2:POKE 2,108:POKE 3,147
70 POKE 66,1:POKE 548,215:POKE 549,145:POKE 66,0
100 DATA 104,201,3,208,67,104,208,64,104,233,1,201,8,176,57,168,10
110 DATA 10,10,10,10,170,104,133,211,104,133,210,104,208,41,104
120 DATA 240,34,201,33,176,34,153,100,146,168,136,177,210,132,212
130 DATA 160,191,209,121,240,5,136,192,255,208,247,152,157,108
140 DATA 146,232,164,212,208,231,96,153,100,146,96,76,46,185
150 DATA 104,201,1,208,248,104,141,244,145,104,141,245,145,96
160 DATA 166,0,16,83,173,220,2,142,220,2,201,145,208,17,173
170 DATA 244,145,48,111,133,189,173,245,145,133,188,76,50,185
180 DATA 255,255,162,4,201,81,240,26,162,0,201,17,240,20
190 DATA 172,31,208,185,92,146,170,240,78,173,15,210,41,8
200 DATA 208,4,138,9,4,170,189,100,146,240,52,169,142,141
210 DATA 8,2,134,0,138,10,10,10,10,10,24,125,100,146,133,7
220 DATA 172,252,2,200,208,37,224,127,240,15,198,7,164,7,185
230 DATA 108,146,141,252,2,152,41,31,208,18,173,31,208,162,127
240 DATA 201,7,208,2,162,255,134,0,169,25,141,8,2,76,138,194
250 DATA 0,0,0,3,0,2,1,0,0,0,0,0,0,0,0,0
300 DATA 169,144,133,106,32,148,239,230,66,169,215,141,36,2,169,145,141,37,2,198
,66,96
```



HA ÉRDEKLI A SZÁMÍTÓGÉP – FÖLTÉTLENÜL JÖJJÖN EL!
DECEMBER 12–13-ÁN 9-TŐL ESTE 7-IG
A MŰSZAKI EGYETEM KÖZPONTI ÉPÜLETÉNEK AULÁJÁBAN
BUDAPEST XI., MŰEGYETEM RAKPART 3.

A tavalyi és tavalyelőtti sikeren felbuzdulva idén is megrendezzük év végi zsibvásárunkat. Tavaly voltunk vagy ötezren, idén megcélozzuk az 5001-et. **Az 5001-ik fizető vendéget egy karácsonyi BITLET-kosárral jutalmazzuk!**

A helyszín a budapestieknek közismert, a vidékieknek segítségképpen: A Keletitől a 7-es busz, a Nyugatitól a 12-es busz, a Délitől a 18-as villamos, az Engels téri buszpályaudvartól az 1-es busz a legjobb közlekedési eszköz. Autósoknak előmelegített parkolóhelyek!

(Vállalkozó kedvűek jöhetnek futva, úszva, esetleg gyalog.)

A KAPUNÁL VÁLASZTÉKOS MODORÚ JEGYSZEDŐINKNÉL DIÁKOK ÉS KATONÁK 15, MÁSOK 25 FORINTÉRT VEHETNEK BELÉPŐT.

PROGRAMCSEREBERE

Idén ismét **50 géphelyet** állítunk föl. Szokás szerint adjuk a tévét és a csatlakozási lehetőséget. A többit önnek kell hoznia! **Egy asztal egy órára 30 forint.** A bérletek előjegyezhetők. Aki december 7-ig befizeti a bérleti díjat, az 20%-kal kevesebbet fizet!

A helyfoglalást telefonon is, személyesen is, postán is lebonyolíthatják. A Csokonai Művelődési Házban reggel 9-től este 9-ig várja jelentkezésüket, vagy hívásukat **Sperber László.**

Telefonok: 690-495 vagy 892-240.
Postacím: Csokonai Művelődési Ház 1153 Budapest XV., Eötvös u. 64–66.

PROGRAMBÖRZE

Aki nem csereberélni akar, hanem saját készítésű programjait, hardver eszközeit kívánja árusítani, az külön megállapodást köthet a rendezőkkel. Az eladni kívánt áru értékétől, a szükséges helytől, technikától függő bérleti díjat kell fizetnie.

HIRDETÉSEK

Idén is rendelkezésükre állnak a helyszínen hirdetési eszközeink. Csereajánlatok, számítógépes barátok keresésére vonatkozó közlemények közzététele jutányos áron, egyéb hirdetések, másolatok a triplájáért.

HANGOS REKLÁM: 10 forintért többször bemondjuk az ön által megadott szöveget

RÖPCÉDULA: A4-es laponként 2 forintért sokszorosítjuk az ön által megadott szöveget. Esetleg a terjesztésben is segítünk.

BEMUTATÓK

Erről egyelőre keveset árulunk el, mert keveset tudunk. Idén is lesz nagyméretű kivetítő, amire bizonnyal fölkerül az **Amiga, a Macintosh, az Atari ST., az Enterprise** képe. Szeretnénk bemutatni **lézerprintert, tervezői munkaállomást,** az **új IBM** család egyik-másik tagját. Ötleteket, bemutatásra érdemes programokat, termékeket szívesen fogadunk. Kérjük, hogy **ötleteikkel keressék meg telefonon Angyalosi Lászlót. (408-603)**

PROFIKNAK

Idén külön meghívót küldünk egy sor IBM kompatíbilis géppel rendelkező cégnek. A „Profi szekció"-ban reményeink szerint üzletkötésekre is sor kerül majd. Természetesen a megkülönböztetés nem jelenti azt, hogy az amatőröket elzárnánk az IBM kompatíbilis, és más professzionális gépekre készült szoftverek, hardverek bemutatójától.

EGYEBEK

Egyebek közt ott lesznek a BIT-LET szaktanácsadói, a bemutatókon alkalmuk lesz kérdezni a különböző gépekhez értő szakemberektől, lesz **JÁTÉKSAROK,** ahol sok-sok számítógépes játék kezelésére vonatkozó tanáccsal látják majd el az érdeklődőket.

És aki nem vette meg még számítógépes barátai, családtagjai részére a karácsonyi ajándékot, annak rendelkezésére állnak a különböző cégek könyveket, programokat árusító pavilonjai!

RENDEZŐK, VÉDNÖKÖK

Az idei BIT-LET Karácsony rendezője a **Csokonai Művelődési Ház** és a **BIT-LET Szerkesztősége,** védnöke az **AGROBANK RT.,** a **KISZ KB.,** a **NOVOTRADE RT.,** a **SZÁMALK,** a **SZÜV COMPUTER M.,** és az **ÁPISZ.** Ha jól érzik magukat, őket dicsérjék, ha rosszul, akkor a rendezőket szidják!

ötlet

SORVEZETŐ

A B C D E F G H I

LESZ-E VALAHA IS REND?

A sorvezető legutóbbi nekifutása (1987. március–július) alkalmából a „rendezésnek" volt a legjelentősebb visszhangja. Ketten(!) is reagáltak magára a Sorvezetőre, sőt az ott kitűzött – nem egyszerű – feladatra (mely egyben a NYERŐ pályázat feladat is volt) 21 MEGOLDÁS érkezett (melyek közül 12 helyes volt). Ezekről részletesebben a pályázat értékelésénél olvashatnak.

Idézzük fel először a feladatot: N darab elemet (számot, nevet stb.) kell nagyság szerint sorbarendezni. Az idézett számban (1987. június 25.) szerepelt két ún. lassú megoldás. A relatív maximum keresés, illetve a rokon „buborék" módszer műveletigénye egyaránt N*N/2, azaz nagyjából ennyi alkalommal kell összehasonlítást végeznünk a rendezés során. Ez ezer adat esetén közel félmillió hasonlítás, ami már percekig is eltarthat.

Ugyanebben a részben előkészítettünk két jóval gyorsabb változatot. A felezéssel történő beszúrás (3.b. megoldás), illetve az összefésülés (4. program) egyaránt általánosítható olyan programmá, ami „csak" N*LOG(N) összehasonlítást végez mindössze. Ez ezer adatnál például félmillió helyett mindössze tízezer (ötvenedrésze!) művelet. Ezt tűztük ki feladatul, egy egyszerűbb speciális esetben (N=128). **Kajári Gábor (Cegléd)** az összefésülésen alapuló programján kívül egy másik rendező-programot is beküldött, mely sok esetben kb. kétszer gyorsabbnak látszik (s mindenesetre rövidebb) az összefésülésen alapuló programnál. Ez a „sok esetben" kb. azt jelenti, hogy ha az adatok eléggé össze vannak keverve (pl. a programban látható véletlenül előállított sorozat ilyen), akkor valóban gyors. Azonban ha pl. kipróbáljuk a programot egy 100 hosszú 1–0 sorozattal, akkor a futási idő kb. két és félszer akkora lesz, mint 100 véletlen adatra (18 másodperc helyett 46 másodperc). A program listáját triviális módosításokkal közöljük.

Ígéretünkhöz híven igyekszünk valami aprósággal honorálni olvasónk nem jelentéktelen fáradozását: magam részéről szívesen megküldöm az áhított Lovász–Gács: Algoritmusok c. könyvét (melyet a közelmúltban adott újra ki a Műszaki Könyvkiadó), másrészt pedig természetesen részt vesz a NYERŐ pályázat sorsolásán.

Ceglédi barátunk levelében elég sokat foglalkozik a könyvekkel. Ehhez kapcsolódva adnék hírt egy újabb jó kiadványról. Az OMIKK jelentette meg a „Számítástechnika középfokon" című munkát. Több mint tíz oldal foglalkozik a rendezésekkel, elméleti kérdésekkel is, és az algoritmusok verbális leírása is szerepel. Ugyancsak a közeljövőben jelenik meg E. Knuth: A programozás művészete c. műve a Műszaki Kiadónál, ami hasonló kérdések alapkönyvének tekinthető.

A RENDEZHETŐSÉGRŐL

Mostani alkalommal nem annyira a rendezés mikéntjéről lenne szó, inkább a rendezhetőségről. Bizony, nem magától értetődő, hogy egy halmaz (tökéletesen) rendezhető! Az eddig tárgyalt esetekben olyan (jól mérhető) szempontok szerint kellett rangsorolni, hogy két dolog összehasonlításával egyértelműen eldönthető volt a köztük lévő reláció (<, >, vagy =). Már a testmagasság is olyan, hogy a mérés (összehasonlítás) eredménye sok mindentől függhet (kihúzza-e magát, fáradt-e stb.). Még sokkal inkább ilyen például a sportverseny területe. Mikor merjük kijelenteni, hogy egy sportoló (egy csapat) jobb a másiknál? Hiába veri meg ellenfelét egyszer, ha legközelebb revansot vesz a másik! Figyelembe vegyük-e, hogy az egyik eredmény fölényesebb volt, mint a másik? Vagy mit kezdhetünk avval a szituációval, amikor A legyőzi B-t, B diadalmaskodik C felett, és A kikap C-től. Melyikük a legjobb, illetve a legrosszabb?

Sok sportágban hosszú idő alatt alakult ki a legjobbak rangsorolásának mikéntje. Gyakran a sportág legnagyobb versenye (olimpia, VB) az alapja egy erősorrendnek. Ez nyilván függ a szerencsétől (egyszeri formaidőzítés, kieséses rendszernél számít a sorsolás stb.). Valamivel jobb a pontverseny: sok verseny eredményét pontozzák, és a pontszámok összege dönt. Ennek hátránya, hogy aki valami miatt kevesebbszer indul, az hátrányban van.

Két sportágat emelnék ki, elsősorban a sakkot (Élő-pontok), másrészt a teniszt (ATP-lista). Nem ismerek más területet (persze ettől lehet), ahol ennyire következetesen történne a rangsorolás. Mindkét rendszer „folyamatos", állandóan a megelőző listát módosítják az újabb eredmények alapján. Sakkban sokkal statikusabban, azaz hiába veri meg valaki az addig élen állót, ettől csak emelkedik, de nem lesz rögtön első! Minden versenyzőhöz minden időpontban hozzá van rendelve egy pontszám. Ezek alapján minden versenynek van egy értéke (minél több jó játékos vesz részt, annál magasabb), és eszerint kap (vagy veszít) pontokat minden résztvevő – helyezése (győzelmeinek száma) szerint. Ily módon szinte minden „esemény" után azonnal kész az új lista. Ezen zseniális rendszer tehát még arra is tekintettel van, milyen körülmények között (milyen erős versenyen) születik egy eredmény. Még egyszer hangsúlyozzuk, hogy nagyon erős szerepe van a múltnak: egy ifjú üstökösnek az élmezőny sok tagját kell megvernie, hogy az elsők közé kerüljön. Mindenesetre nem meglepő, hogy Élő Árpád professzor – aki a sakkozók rangsorolásának rendszerét kidolgozta –, matematikusként is ismert, hiszen elméletileg nagyon jól megalapozott rendszert alkotott – de statikussága miatt igazából csak sakkra ideális.

A teniszezőknél is tekintettel vannak régebbi eredményekre, de kevésbé. Erre a sportágra ugyanis sokkal jellemzőbb a hirtelen berobbanás (McEnroe, Becker stb.). Minden nagyobb versenyért lehet pontokat kapni (ha elöl végez az illető), figyelembe veszik, ki hány versenyen indult, nézik a győzelmek/vereségek arányát stb. Elég bonyolult

```
800 INPUT N
840 DIM A(N)
900 FOR I=1 TO N:A(I)=INT(100*RND(0)):NEX
TI
910 FOR I=1 TO N:PRINT A(I),:NEXTI
1000 REM *** GYORSRENDEZO ***
1050 P=0
1060 F(0)=1:L(0)=N
1070 F=F(P):L=L(P)
1080 P=P-1
1090 GOSUB 1210:REM FELOSZTO HIVASA
1100 IF L-I>J-F THEN 1130
1110 IF I<L THEN P=P+1:F(P)=I:L(P)=L
1120 L=J:GOTO 1150
1130 IF F<J THEN P=P+1:F(P)=F:L(P)=J
1140 F=I
1150 IF L>F THEN 1090
1160 IF P>-1 THEN 1070
1167 PRINT
1170 FOR I=1 TO N:PRINT A(I),:NEXT I
1189 END
1190 REM *** FELOSZTO SZUBRUTIN ***
1210 I=F:J=L
1220 K=INT((F+L)/2):M=A(K)
1230 IF A(I)<M THEN IF I<=L THEN I=I+1:GO
TO 1230
1240 IF A(J)>=M THEN IF J>=F THEN J=J-1:G
OTO 1240
1250 IF I>=J THEN 1280
1260 C=A(I):A(I)=A(J):A(J)=C
1270 I=I+1:J=J-1
1280 IF I<=J THEN 1230
1290 IF I<>F THEN 1330
1300 IF L<=F+1 THEN 1320
1310 C=A(I):A(I)=A(K):A(K)=C
1320 I=I+1
1330 RETURN:REM VISSZATERES A FELOSZTOROL
```

összeállítani a rangsort, jellemző, hogy már sok évvel ezelőtt is számítógéppel végezték (pedig akkor ez még nem volt kötelező divat).

MIKOR JÓ EGY RANGSOR?

Ezek után nézzük meg, hogy mikor jó egy rangsor? Elemi követelmény, hogy ha A legyőzte B-t, akkor előzze is meg. De mi van, ha A háromszor nyert, B pedig két alkalommal? Erre azért nyugodtan rámondhatjuk, hogy A (valamivel) jobb. Mit mondjunk, ha mindketten 2-2 ízben győztek? Nyugodtan állíthatjuk, hogy (nagyjából) egyformák, döntsön a mások elleni mérlegük. Persze a holtversenyt nem kell kizárni, csak ne vigyük túlzásba! Sajnos a fenti A–B–C körbeverés mindenképpen ellentmondást eredményez a listában. Nem szerencsés ilyenkor azt mondani, hogy ők „egyformák". Gondoljunk arra, hogy az összes eredmény alapján A az első, B a hetedik, C a nyolcadik. Ha a „C pedig megverte A-t" ellentmondást A,B,C egyformaságával próbálnánk feloldani, akkor hozzájuk kellene csapni a köztük lévő öt versenyzőt: 1–8-ig mindenki egyforma! Ezért kell csínján bánni a holtversennyel.

Ezek alapján talán nem meglepő, hogy javaslom tökéletesnek mondani azt a rangsort, ami nem tartalmaz „ellentmondást". Persze ez általában lehetetlen: azok a fránya játékosok összevissza ingadoznak formailag. Mondjuk inkább azt, hogy egy lista annál jobb, minél kevesebb az ellentmondás. Érdemes egy picit elgondolkodni azon, hogyan (milyen algoritmus szerint) lehetne sorbarendezni versenyzőket.

Még csak annyit, hogy néhány ilyen módszer közül az egyikkel megvizsgáltam (annak idején) az 1979. évi ATP ranglistát. Az újságokban közölt listában hemzsegtek az ellentmondások, persze ez főleg a múlt figyelembe vétele: McEnroe 79-ben már rendesen verte Connorst, de előző évben még sehol sem lévén, mögötte volt!

Inkább a mérkőzések eredményeit boncolgattam, és a következő adódott: N jelenti, hogy az ATP lista első hány emberének mérkőzéseit vettem figyelembe.

játékos (N)	6	8	10	12	14	20
mérkőzés N(N–1)/2	15	28	45	66	91	190
ellentmondások	1	1	4	7	11	19

Ha ezek után valaki erkölcstelen bundázóknak tartja a profik színe-javát, azt megnyugtatom, hogy RND-vel generált „mérkőzések" esetén sokszor ennyi ellentmondást lehet csinálni.

KÖVETKEZETLEN ÍZLÉSÜNK

Arra, hogy nemcsak a teniszvilág „következetlen", hanem például a kedves olvasó is, hadd mondjak egy pár éve igen népszerű példát. Gondoljunk néhány (N) gyümölcsre. Képezzünk ezek közül minden párt (N*(N—1)/2 darab), és mondjunk véleményt minden párról: melyiket kedveljük jobban. Ha nem figyelünk, hanem kizárólag az emlékeinkre, az ízlésünkre támaszkodunk, akkor már N=12–16 esetén ellentmondásba keveredünk. N=25 esetén kiváló eredmény, ha valaki tíznél kevesebbet hibázik! Természetesen lehet „csalni": magamban sorbaállítom a dolgokat, és mindig ennek megfelelően mondok véleményt...

A REND IS RENDEZHETETLEN

Végül szerepeljen egy példa, hogy nemcsak a versenyzők formaingadozása felelős a rendetlenségért. Tekintsünk 9 GO játékost jelentős tudáskülönbséggel. A GO egyik fő vonzereje, hogy egy jobb játékos szinte biztosan megver egy rosszabbat, tehát feltehetjük, hogy a 9 versenyzőnk szinte biztosan rangsorolható, és senki nem képes megverni előtte állót.

Csináljunk három csapatot belőlük a következőképpen:
1. csapat 1. (legjobb), 6., 8.
2. csapat 3., 5. 7.
3. csapat 2., 4., 9. (utolsó)

A három csapat körmérkőzést játszik, melynek során minden csapattag játszik az ellenfél mindhárom versenyzőjével. Mi lesz vajon az eredmények alapján a csapatok sorrendje? Láthatjuk tehát, hogy hiába teremtünk rendet az egyéneknél, csoportok esetén már minden felborulhat!

Török Turul

A NYERŐ-PÁLYÁZAT ÉRTÉKELÉSE

NYERŐ nevű egyfordulós pályázatunkra (rendezés összefésüléssel) 21 megoldás érkezett. Ezek közül néhány hibás, többen pedig nem a kitűzött feladatot oldották meg (azaz nem a két rendezett sorozat összefésülésén alapuló programot írtak, vagy pedig azon alapuló, de nem elég gyors programot).

A helyes megoldás az, hogy először a szomszédos számpárokat rakjuk helyes sorrendbe, majd a szomszédos párok összefésülésével rendezett négyeseket hozunk létre, ezután a szomszédos négyesek összefésülésével rendezett nyolcasokat csinálunk,..., végül a rendezett sorozatot megkapjuk a sorozat két felének (ahol a felső fele lehet hogy kisebb, mint az alsó) az összefésüléséből. Így N adat esetén összesen kb. 3*N*LOG(N) összehasonlítást kell csinálnunk, tehát a program bizonyos értelemben a lehető leggyorsabb lesz, ugyanis N*LOG(N) összehasonlításra mindig szükség van, ahogyan azt Török Turul is írja a Sorvezetőben.

12 helyesen megfejtőnk között a nyereményeket a BIT-LET Karácsonyon, dec. 12-én du. 4 órakor sorsoljuk ki.

A MEGFEJTŐK: Hernyik András, Szentes; Kajári Gábor, Cegléd; dr. Kecskeméti László, Nagykőrös; Makula Tibor, Orosháza; ifj. Nagy Béla, Esztergom; Nagy Béla, Gyula; Pakai Tibor, Szolnok; Szalkai István, Devecser; Szilvási Margit, Mezőkövesd; Timkó Attila, Budapest; Urbán Péter, Budapest; Vindics István, Mecseknádasd.

Megjegyezzük, hogy Pakai Tiboron kívül mindenki 2*N (+ néhány) memóriát használt, egyedül ő tudta a feladatot 1.5*N (+ néhány) memóriával megoldani.

Mi Szilvási Margit programját közöljük, mely Commodore 64-re íródott, de C 16-on is, sőt kisebb változtatásokkal bármilyen gépen futtatható. A program még kicsit gyorsítható, ha az A és B tömbök helyett egy A(N,1) tömböt használunk, s egyszer a nulladik szintről fésülünk össze az elsőre, majd az elsőről a nulladikra stb. A program az elején a párokat nem az összefésülő eljárás segítségével rendezi, mert így gyorsabb. A program sebessége 10 szám esetén 1.6 másodperc, 100 szám esetén 29.5 másodperc, 1000 szám esetén 7 perc 16 másodperc.

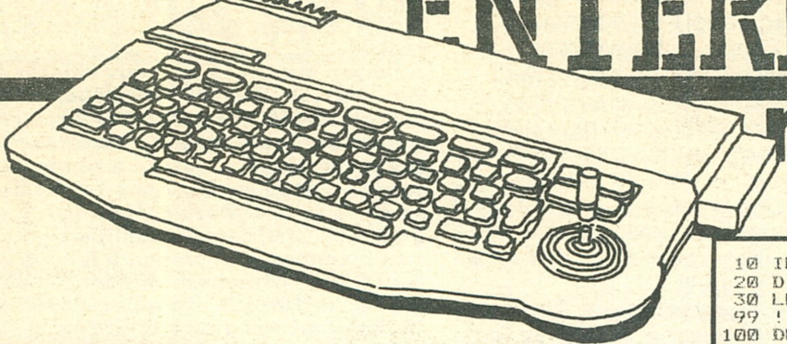
```
10 REM**********************
20 REM*       RENDEZO       *
30 REM**********************
40 GOTO170
50 REM*****  FESULES  *****
60 I=M1:J=M2:K=1
70 IF A(I)>A(J) THEN B(K)=A(J):J=J+1:GOTO 90
80 B(K)=A(I):I=I+1
90 K=K+1:IF I<M1+N1 AND J<M2+N2 THEN 70
100 IF I>=M1+N1 AND J>=M2+N2 THEN 130
110 IF I>=M1+N1 THEN L=M2+N2-J:FOR X=0 TO L:B(K+X)=
 A(J+X):NEXT X:GOTO 130
120 L=M1+N1-I:FOR X=0 TO L:B(K+X)=A(I+X):NEXT X
130 FOR I=M1 TO M1+N1+N2-1
140 A(I)=B(I-M1+1)
150 NEXT I
160 RETURN
170 INPUT N
180 DIM A(N+1),B(N+1)
190 FOR I=1 TO N:A(I)=INT(RND(1)*1000):PRINT A(I),:
 NEXT I:PRINT
200 AA=TI:IF N=1 THEN 380
210 FOR I=1 TO INT(N/2)*2 STEP 2
220 IF A(I)>A(I+1) THEN B=A(I):A(I)=A(I+1):A(I+1)=B
230 NEXT I
240 LL=1:I=2
250 IF N<=I THEN 270
260 LL=LL+1:I=2*I:GOTO 250
270 L0=1
280 FOR KK=1 TO LL-1
290 L0=2*L0
300 FOR T=1 TO N STEP 2*L0
310 M1=T:N1=L0
320 M2=T+L0:N2=(L0+N-M2+1-ABS(L0-N+M2-1))/2
330 IF N2<=0 THEN 350
340 GOSUB 50
350 NEXT T
360 NEXT KK
370 AA=TI-AA
380 FOR I=1 TO N:PRINT A(I),:NEXT I:PRINT
390 PRINT AA/60;" SEC"
400 END
```

ATARI NYERŐ

A pályázaton több mint 200-an indultak, 139-en küldtek mindhárom feladatra „nem teljesen rossz" megoldást. Mivel a 139*3 megoldást részletesen végigbogarászni reménytelenül nagy feladatot jelentene, ezért a sorsolás, mely a BIT-LET Karácsonyon, dec. 12-én du. 4 órakor lesz, úgy zajlik le, hogy a 139 pályázatból kihúzunk egyet, s a helyszínen végezzük el a részletes kiértékelést, ha a 3 megoldás valamelyike rossz, akkor újra húzunk.

CENTRUM

ENTERPRISE® nyerő



A 2. FELADAT MEGOLDÁSA

A feladat az volt, hogy egy olyan programot írjunk, mely az ismertetett egyszemélyes játékot helyesen végigjátssza. Az itt közölt program az ENTERPRISE IS-BASIC-jében íródott.

A program rövid leírása:

Ha játsszuk a játékot egy darabig, hamar megfigyelhetjük, hogy a feladat teljesítése két részből áll:

• először azt kell elérnünk, hogy a világos és sötét korongok váltakozva legyenek, s a bal szélen már világos korong legyen (vagy a bal szélen egy üres hely, s mellette egy világos korong),

•• innen pedig az eddig csináltaknak majdnem pontosan a fordítottjával elérhetjük, hogy minden a helyére kerüljön.

A program ezen az elven működik, s még használja a következő észrevételt:

Tegyük fel, hogy az először megkívánt helyzetet egy eggyel kisebb méretű feladatra már megoldottuk (a belső N–1 sötét és a belső N–1 világos korong már megfelelően váltakozva van). Innen az N méretű feladatra megkívánt helyzetet már egy egyszerű lépéssel és egy egyszerű, ciklussal megszervezhető ugrássorozattal elérhetjük – csak N páros vagy páratlan voltától függően kell egy picit mást csinálni.

A játék tábláját a **T** tömbben tároljuk Az **UGRAS** nevű eljárás a **POZ** helyen lévő koronggal lép vagy ugrik a megfelelő irányba (az irányt a **POZ** helyen lévő korong színe határozza meg), annak megfelelően, hogy hol van az üres hely. (Az algoritmus gondoskodik róla, hogy az illető koronggal lehessen lépni, így itt vizsgálat nem kell.) A **KIIR** eljárás kiírja a tábla pillanatnyi helyzetét, ezt minden lépés után meghívjuk. A **BEFELE** rekurzív eljárás gondoskodik az előbb említett első fázisról, M=1 esetén (1 világos, 1 sötét korong) a feladat egyszerű, egyébként először meg kell oldani M–1-re, majd a szabad szélsővel (melynek helye a középtől M mezőre van M párosságától függő irányban) kell egyet lépni, s ezután a másik szín bábuival ugrani egyet-egyet. A **KIFELE** eljárás hasonlít a **BEFELE**-hez, csak ez a második fázis végrehajtásáról gondoskodik.

Ezután a főprogram roppant egyszerű (s strukturált programoknál ez így van rendjén), csak a kezdőállapotot kell létrehozni és kiírni, majd meghívni egymás után a **BEFELE** és **KIFELE** eljárásokat.

Aki most találkozik először rekurzív eljáráshívással, annak érdemes a program futását pl. N=4-re végigkövetni.

```
 10 INPUT N
 20 DIM T(2*N+2)
 30 LET S$="+ o"
 99 !
100 DEF UGRAS(POZ)
110   LET FIG=T(POZ)
120   LET T(POZ)=0
130   LET UG=2+(T(POZ+FIG)=0)
140   LET T(POZ+UG*FIG)=FIG
150   CALL KIIR
190 END DEF
199 !
200 DEF KIIR
210   FOR J=1 TO 2*N+1
220     PRINT S$(T(J)+2:T(J)+2);
230   NEXT J
240   PRINT
250 END DEF
299 !
300 DEF BEFELE(M)
310   IF M=1 THEN
320     CALL UGRAS(N)
330     CALL UGRAS(N+2)
340   ELSE
350     CALL BEFELE(M-1)
360     LET PAR=2*(M<>2*INT(M/2))+1
370     LET KP=N+1+M*PAR
380     CALL UGRAS(KP)
390     FOR I=1 TO M
400       CALL UGRAS(KP-I*2*PAR)
410     NEXT I
420   END IF
430 END DEF
499 !
500 DEF KIFELE(M)
510   IF M=1 THEN
520     CALL UGRAS(N+1)
530   ELSE
540     LET PAR=2*(M=2*INT(M/2))+1
550     LET KP=N+1+(M-1)*PAR
560     CALL UGRAS(KP)
570     FOR I=1 TO M-1
580       CALL UGRAS(KP-I*2*PAR)
590     NEXT I
600     CALL KIFELE(M-1)
610   END IF
620 END DEF
699 !
700 FOR I=1 TO N
710   LET T(I)=1
720   LET T(I+N+1)=-1
730 NEXT I
740 LET T(N+1)=0
750 CALL KIIR
760 CALL BEFELE(N)
770 CALL KIFELE(N)
780 END
```

4. FELADAT:

A feladat szövegfeldolgozás lesz. Tegyük fel, hogy egy hosszú magyar nyelvű szövegben meg akarjuk számolni, hogy az egyes kötőszavakból (és, s, de, hogy, vagy, azonban, ezért, így, meg, sőt) hány db van.

A programot az ENTERPRISE IS–BASIC-jében kell megírni, strukturáltan, olvashatóan. A szöveget egy string-tömbben tároljuk, melynek elemei 254 karakter hosszúak (ugyanis 254 egy string maximális hossza), s egy-egy ilyen string szavain ne ciklusszervezéssel, hanem rekurzív eljáráshívással (példát a múltkori számunkban, s a mostani megoldásunkban láthatunk rá) menjünk végig. Egy szó átnyúlhat egyik stringből a következőbe! Vigyázzunk, egy szó végét nem csak space, hanem vessző, pontosvessző, kettőspont, pont, felkiáltójel vagy kérdőjel is jelezheti. Használjuk az előző számunkban ismertetett POS függvényt! A végén program írja ki, hogy a fent említett kötőszavak közül melyikből hány db van.

A programot papíron kérjük beküldeni a működést ismertető leírással együtt.

ENTERPRISE nyerő



Kérjüklevágni és a levélre felragasztani! Beküldési határidő: 1988. január 15.

BIT-LET Szerkesztőségünk címe: ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál. Tördelő: Domokos Imre. Hírrovat: Csányi György. Gépnyerő: Király Zoltán. Posta: Halász Péter.